

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SYSTÉM PRO EVIDENCI CHOVŮ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RADKA ENTLOVÁ

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

SYSTÉM PRO EVIDENCI CHOVŮ

THE BREEDING EVIDENCE SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

RADKA ENTLOVÁ

VEDOUCÍ PRÁCE
SUPERVISOR

ING. OTA JIRÁK

BRNO 2010

Abstrakt

Bakalářská práce se zabývá návrhem a vývojem webové aplikace pro chovatele. Věnuje se problematice plemenných chovů a požadavkům pro jejich chov. Tato aplikace nabízí uživatelům členství ve skupinách a možnost být v kontaktu s přáteli nebo najít nové. Popisuje implementaci aplikace vytvořené pomocí architektury MVC. Aplikace byla vytvořena pomocí JavaServer Pages, Struts2 a AJAX.

Abstract

The bachelor thesis deals with design and development of a web application for breeders. It focuses on problems of racial breed and their requirements for the breeding. This application provides users group memberships and the opportunity to be in touch with friends or find new ones. It describes the implementation of the application built using MVC architecture. The application was developed using JavaServer Pages, Struts2 and AJAX.

Klíčová slova

Webová aplikace, informační systém, chov zvířat, plemena, evidence chovu, JSP, Struts 2, AJAX, HTML, JavaScript, CSS

Keywords

Web application, information system, breeding, breed, breeding evidence, JSP, Struts 2, AJAX, HTML, JavaScript, CSS

Citace

Entlová Radka: Systém pro evidenci chovů, bakalářská práce, Brno, FIT VUT v Brně, 2010

Systém pro evidenci chovů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením Ing. Oty Jiráka.

Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

.....

Radka Entlová

14. května 2010

Poděkování

Zde bych ráda poděkovala své rodině, příteli a kamarádům, kteří mě ve studiu podporovali.

Speciální poděkování patří vedoucímu Ing. Otu Jirákovi a konzultantovi Ing. Petru Chmelařovi za podnětné návrhy a odbornou pomoc při řešení problémů, které se během řešení této bakalářské práce vyskytly.

© Radka Entlová, 2010

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

Obsah.....	1
1 Úvod.....	3
1.1 Člověk a zvíře.....	3
2 Chov plemen	5
2.1 Základy genetiky	5
2.1.1 Molekulární základy dědičnosti	5
2.1.2 Genetika eukariotické buňky	8
2.1.3 Dědičnost mnohobuněčných organismů	9
2.1.4 Genetika psa.....	10
2.1.5 Mutace	12
2.2 Základy šlechtění plemen.....	12
2.2.1 Selektce	12
2.2.2 Metody plemenitby	13
3 Technologie.....	16
3.1 MVC model.....	16
3.2 JavaServer pages	17
3.3 Struts 2	18
3.4 Ajax	20
3.5 HTML, CSS, Javascript	20
4 Návrh aplikace	22
4.1 Existující systémy pro evidenci chovu	22
4.2 Požadavky na systém	22
4.3 Use case diagram.....	24
4.4 ER diagram.....	25
5 Implementace	26
5.1 Session.....	26
5.2 Profil uživatele	27
5.3 Zvířata	28
5.4 Zprávy	29
5.5 Přátelé.....	31
5.6 AJAX.....	32
5.7 View	32
5.8 Problémy při tvorbě aplikace	32
6 Testování.....	33

7	Závěr	36
	Literatura	37
	Seznam příloh	39
	Příloha 1.....	40
	Příloha 2.....	41
	Příloha 3.....	46

1 Úvod

Tématem mé bakalářské práce je systém pro evidenci chovů. Informační systém bude ve formě webové aplikace. V dnešní době svět informačních technologií stále více proniká do našeho každodenního života. Existuje nespočet různých webových aplikací, která usnadňují lidem život, ať už se jedná o internetové obchody či internetové bankovníctví. O to se snaží i tato aplikace, jejím cílem je dosáhnout informačního systému, který by pomáhal nejen chovatelům holubů a psů. Systém ve formě webové aplikace je přístupnější a umožňuje chovatelům snadnější a přehlednější komunikaci mezi sebou. Uživatelům bude nabídnuta možnost navazovat přátelství a vyhledat vhodné jedince pro své chovy. Systém by měl pomoci uživatelům při vyhledávání v rodokmenech jednotlivých jedinců. Pro velký věkový rozsah uživatelů je kladen důraz na jednoduchost.

Práci jsem rozdělila do sedmi kapitol. První a poslední kapitolou je úvod a závěr. Poté popisují základy chovu a šlechtění plemen. Jsou zmíněny i základy genetiky a různé problémy, které se mohou vyskytnout při křížení. V další kapitole se zabývám technologiemi, které jsem použila při vytváření tohoto informačního systému. Je zde popsána JavaServer Pages, Struts 2, AJAX a také technologie pro úpravu vzhledu. Ve čtvrté kapitole je návrh a specifikace aplikace, je zde uveden diagram případů užití. V páté kapitole se věnuji postupu při implementaci jednotlivých stránek. Další menší kapitolou je testování aplikace.

1.1 Člověk a zvíře

Přibližně před 10 000 lety se k člověku přidal jeho věrný přítel – pes [21]. Tímto začala postupná domestikace zvířat. Pod pojmem domestikace nebo také zdomácnění, či ochočení, si můžeme představit postupnou přeměnu divokého druhu na druh, který můžeme chovat bez většího nebezpečí v zajetí. Za domestikované zvíře je považováno takové, které je chováno 30 let nebo alespoň 30 generací. V posledních letech se jedná hlavně o domestikaci kožešinové zvěře, akvariálních ryb aj. Ačkoliv si pod domestikací představíme většinou domestikaci zvěře, lze domestikovat i jiné živé organismy, např. rostliny. Lidé postupem času začali chovat zvířata pro různé účely. Jedním z největších důvodů byla obživa. Zvířata nesloužila ale člověku jen jako zdroj jídla, později se pro člověka stala zvířata nepostradatelnou součástí života. Pomáhala mu překonávat vzdálenosti, nosit břemena a přepravovat těžký náklad, obhospodařovat půdu a nebo lovit kořist, hlídat majetek, doručovat první poštovní psaní, a tak dále. Stala se jeho společníkem. Člověk pro ně našel spoustu rozličných využití, která přetrvala do současné doby. V dnešní uspěchané době jsou zvířata stále nedílnou součástí našeho života. Jsou stále nenahraditelní jako naši dlouhověcí společníci a pomocníci. I když některé jsme sice vyměnili za traktory a nebo poplašná zařízení, koně už nám neslouží jako hlavní dopravní prostředek a holubi už nelétají s dopisy. Stále je potřebujeme,

ať už je využíváme jako zdroj potravin, surovin pro výrobu oblečení aj., tato zvířata stojí spíše v pozadí, převážně se spíše v našich životech objevují zdomácnělá zvířata v rolích věrných společníků v době, kdy lehce ztrácíme lidské kontakty, nebo jako stálí pomocníci, jakými jsou například asistenční psi pro tělesně či mentálně postižené lidi, nebo psi policejní. Proto mají domácí zvířata v této moderní, hektické a často i stresující době, době plné různých technických pokroků a objevů, svoje místo po boku člověka a tvoří důležitou část našeho světa.

2 Chov plemen

V této kapitole jsou hlavním tématem zvířata, také se zde dočtete o základech genetiky, principech šlechtění a způsobu chovu domácích zvířat. Je zde zmíněno i několik problémů, které se mohou objevit při vzájemném křížení. Většinu teorie vysvětluji na psech, protože jeho genom byl jako jeden z mála genomů ze zvířecí říše celý rozluštěn. V této kapitole jsem čerpala především z těchto dvou knížek Biologie pro gymnázia [1] a Genetika a šlechtění plemen psů [2].

2.1 Základy genetiky

Za zakladatele genetiky je považován Johann Gregor Mendel, který se narodil roku 1822 v Hymnicích na Moravě. Je znám svými pokusy s křížením rostlin, kterými strávil 10 let života, zkřížil více než 30 tis. rostlin. Vše statisticky a matematicky zpracoval, zdokumentoval a vytvořil základní pravidla dědičnosti. Za svého života však nebyl pochopen, jeho práce došla uznání až 16 let po jeho smrti. Znění Mendlových zákonů můžete nalézt na <http://genetika.wz.cz/dedicnost.htm>.

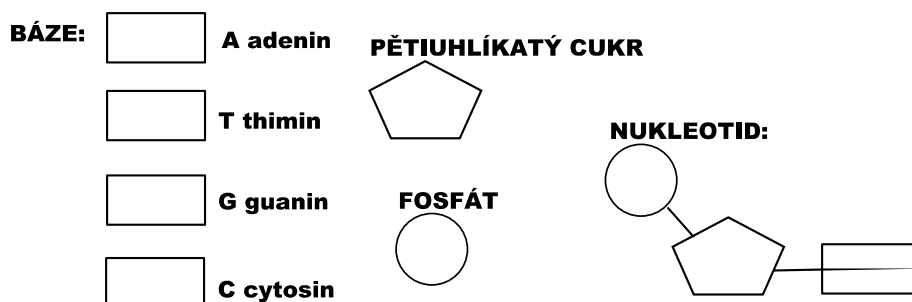
Existují dva typy buněk, prokaryotní a eukaryotní. Rozdíl mezi těmito buňkami je velký. Prokaryotické buňky tvoří pouze jednobuněčné organismy, nikdy netvoří různé tkáně. Zástupci jsou prvobuněční živočichové, kterými jsou bakterie, sinice apod. Ostatní živé organismy včetně rostlin a hub patří do eukaryot. Obě buňky jsou jednoznačně rozlišitelné.

Genetika je věda, která zkoumá dědičnost a proměnlivost zkoumaného subjektu. S dědičností a proměnlivostí souvisí rozmnožování, což je schopnost organismů vytvářet novou generaci. Tato nová generace obvykle zdědí jejich vlastnosti. Rozmnožování se dělí na nepohlavní a pohlavní. Nepohlavním rozmnožováním vznikne nový jedinec klonováním původního organismu. Stačí k tomu například orgán rodičovského organismu. Po splynutí haploidních ($2n$) buněk při pohlavním rozmnožování vznikne nový jedinec, který má vlastnosti zděděné částečně od matky a částečně od otce. Takto vzniklé generace označujeme za potomstvo daného organismu. Potomstvo dědí znaky rodičů, ty jsou dominantní nebo recesivní. Recesivní znak je takový, který se projeví pouze u 25% potomků. Pokud je vliv alel stejný, jedná se o vztah kodominantní. Alela je přesná forma genu. Znaky jsou řízeny geny. Gen se nachází v buňce a je zastoupen dvěma alelami, jedna pochází ze samčího organismu, druhá ze samičího. Pohlavní buňky nazýváme gamety, každá nese jednu alelu.

2.1.1 Molekulární základy dědičnosti

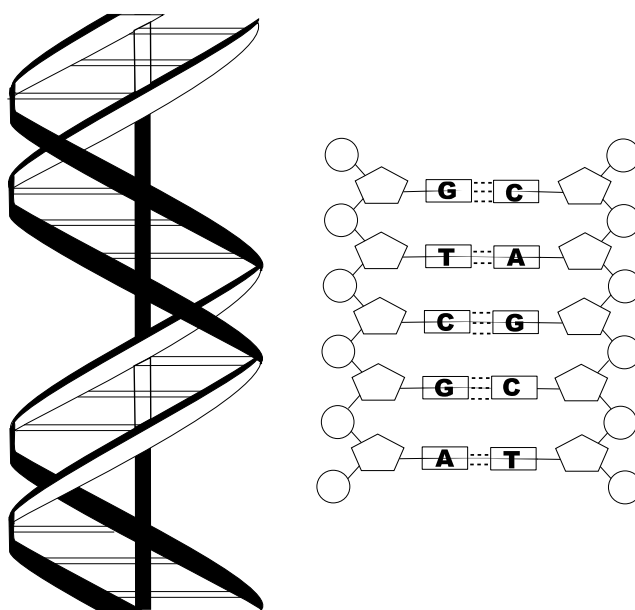
Kyselina deoxyribonukleová (zkratka DNA) je u většiny organismů nositelkou genetických informací, skládá se z nukleotidů. Nukleotid vznikne spojením organické báze, pěti-uhlíkatého cukru

a kyseliny trihydrogenfosforečné (**Obrázek 2.1**). Jsou čtyři druhy organických bází, adenin, thymín, guanin, cytosin. Tyto báze se nacházejí v makromolekule DNA.



Obrázek 2.1: Typy bází a schéma následného spojení v nukleotid (překresleno z [1])

Molekulu DNA tvoří dva polynukleotidové řetězce, ty se společně točí do pravotočivé dvoušroubovice (**Obrázek 2.2**). Vlákna dvoušroubovice jsou k sobě poutány vodíkovými vazbami, tyto vazby jsou mezi bázemi. Vazby vznikají následovně: dvě vodíkové vazby jsou mezi adeninem a thyminem, tři vodíkové vazby mezi guaninem a cytosinem. Tyto dvojice bází označujeme za komplementární. Pořadí nukleotidů je velice podstatné, protože má důležitý vliv na přenos genetické informace. Dvoušroubovice DNA má významnou vlastnost samozdvojení, tedy replikace. Při replikaci vzniká matrice, nebo-li vzor, jedná se o rozvolněné polynukleotické řetězce. Vodíkové vazby se zruší, k tomu přispěje enzym DNA-polymerázy. Na rozdělené polynukleotidové řetězce se navazují volné nukleotidy, které jsou komplementární. Dvě nové dceřiné šroubovice obsahují každá po jednom mateřském řetězci z původní DNA.

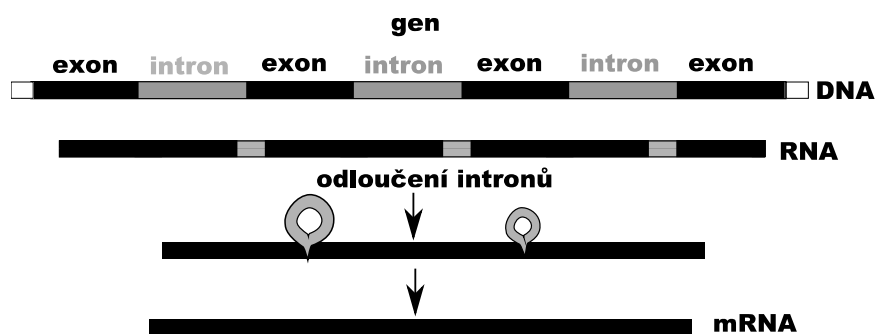


Obrázek 2.2: Schéma dvoušroubovice DNA (překresleno z [1])

Genetické informace jsou uloženy v genetickém kódu v základní struktuře nukleových kyselin. V jednom vlákně makromolekuly DNA můžeme najít uloženo více genů, jsou poskládány za sebou. V buňkách se nachází dvacet druhů aminokyselin, tyto kyseliny syntetizují bílkoviny. Každá z těchto bílkovin je kódována třemi po sobě jdoucími bázemi. Tomuto spojení říkáme triplet nebo kodon. Provedou-li se různé kombinace se čtyřmi bázemi, může vzniknout až 64 kodonů.

Odpovědnost za vznik dědičné vlastnosti nese gen, ten pořadí nukleotidů určuje pořadí aminokyselin v dané bílkovině. Existují tři typy genů - gen strukturní, gen regulátorový, geny pro RNA. Geny bývají většinou uloženy v chromozomech jádra, říkáme tomu jaderný genom. Mimosjaderných genů je méně a jsou uloženy mimo jádro v plastidech a mitochondriích eukaryotických buněk. Při mitóze dochází k přenosu genu mateřské buňky do buněk dceřiných, nazýváme to replikací, nebo-li zdvojením molekuly DNA. Dalším důležitým pojmem je exprese genu. Jedná se o vyjádření genetické informace, která je uložena v makromolekule DNA, do fenotypového znaku. Pokud se bere v úvahu molekulární úroveň, probíhá exprese genu na dvou stupních. Prvním stupněm je transkripce genu, jedná se o tzv. přepis genu. Působením RNA-polymerázy dochází k uvolnění vlákna. Toto uvolnění je dočasné a dochází k němu v místě, kde je uložena genetická informace. Jedno z vláken se stane matricí, ke které se vytvoří komplementární báze mRNA. Nové vlákno uvolněné z matrice se mimo jádro připojí k ribozomu. Druhým stupněm je translace genu, je to překlad genetické informace, která je uložena v přesném pořadí v nukleotidů mRNA, do pořadí aminokyselin v polypeptidické řetězci. Dojde ke spojení kodonů mRNA a antikodonů tRNA, tyto antikodony nesou různé aminokyseliny. Pokud se mezi aminokyselinami vytvoří peptidické vazby, zaniknou vazby mezi kodony a antikodony, tím dojde k uvolnění tRNA. Aminokyseliny jsou vázány na ribozomech do polypeptidického řetězce s primární strukturou. Tato struktura je určena strukturou mRNA.

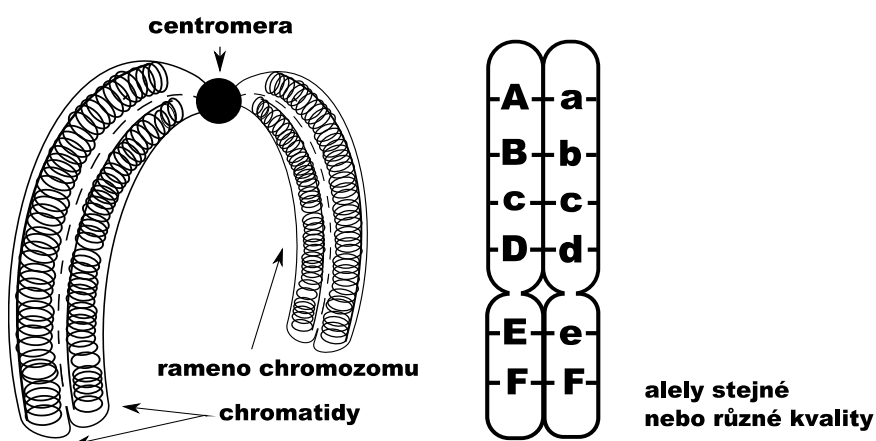
V eukaryotické DNA je gen rozdělen na několik úseků. Těmto úsekům říkáme exony, mezi exony jsou introny. U intronů není zcela jasná funkce, předpokládá se, že má význam v evoluci při vytváření nových vlastností, které nemusí vznikat pouze mutací, ale i vypuštěním intronů a kombinací exonů.



Obrázek 2.3: Schéma přepisu eukaryotické DNA (překresleno z [1])

2.1.2 Genetika eukariotické buňky

V genomu jsou obsaženy všechny genetické informace organismu. Genom je uložen v DNA a zahrnuje všechny geny. Genom je vlastně kompletní sekvence DNA jedné sady chromozomů. Schéma chromozomů je znázorněno na **obrázku 2.4**. Jaderný genom eukaryotické buňky má přesně daný počet chromozomů, ty skládá i z bílkovin. Jednotlivé druhy organismů mají charakteristický, neměnný počet a dokonce tvar chromozomů. Rozlišujeme haploidní (n) a diploidní ($2n$) sadu chromozomů. **Tabulka 2.1** ukazuje přehled počtu chromozomů různých organismů. Haploidní počet obsahují buňky pohlavní, diploidní sadu obsahují buňky tělní, které vznikly dělením zygoty, a samotná jádra zygoty. Zygota je buňka, která má celou sadu chromozomů. Karyotyp je soubor všech chromozomů, které se nacházejí v jádře buněk. Geny jsou vždy uloženy lineárně za sebou. V chromozomu najdeme geny na svých místech, těm říkáme lokusy. Pomocí morganů měříme vzdálenost genů v chromozomu. Abychom mohli určit jednotlivá místa, kde se nachází jednotlivý gen, ukládáme tyto informace do chromozómových map.



Obrázek 2.4: Schéma chromozomu a chromozómového páru (překresleno z [1])

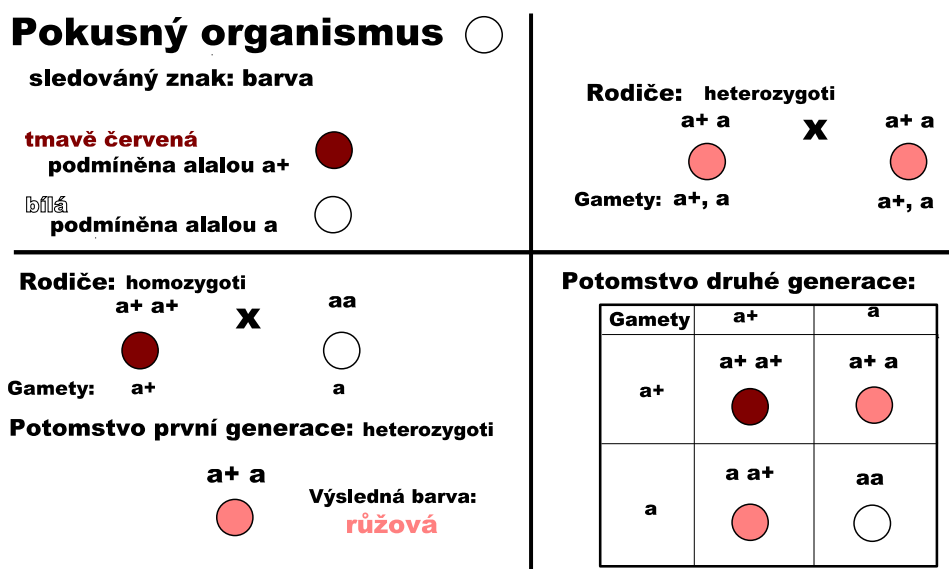
ORGANISMUS	Počet chromozomů	
	2n	n
Brambor	48	24
Hrách	14	12
Kukuřice	20	10
Rajče	24	12
Kůň	64	32
Kapr	104	52
Člověk	46	23

Tabulka 2.1: Přehled počtu chromozomů u různých organismů (převzato z [1])

2.1.3 Dědičnost mnohobuněčných organismů

Při pohlavním rozmnožování jedinec není přímým pokračovatelem ani jednoho ze svých rodičů, samozřejmě pokud to bereme z genetického hlediska. Tento jedinec získá haploidní sady dvou gamet od obou rodičů, v lokusech těchto chromozomů jsou různé alely. Nový jedinec má novou kombinaci genotypu složeného z alel svých rodičů. Z tohoto hlediska je pohlavní rozmnožování na rozdíl od nepohlavního, kdy nový jedinec je přímým pokračovatelem svého rodiče, na vyšším vývojovém stupni. Tato různorodost pomáhá přežít organismům určitého druhu při neustále se měnících podmínkách na této planetě. Díky tomuto je život na Zemi tak pestrý.

Znaky označují jednotlivé vlastnosti daného organismu, dělíme je na morfologické, funkční a psychické. Morfologické znaky jsou znaky popisující tvar a velikost těla, orgánů, buněk. Psychické znaky jsou pouze u člověka. Také můžeme rozdělit znaky na kvalitativní a kvantitativní. Kvalitativní jsou popisující kvality různých vlastností, můžeme si pod tím představit třeba typ krevní skupiny, barvu srsti, aj. Kvantitativní znaky jsou znaky, které můžeme změřit, zvážit, apod. Jedná se například o tělesnou hmotnost nebo výšku. Pokud vezme všechny kvalitativní a kvantitativní znaky určitého jedince daného druhu, tak vzniklý souhrn nazveme fenotypem. Od rodičů jedinec nezíská jednotlivý znak, ale pouze předpoklady. Z molekulárního hlediska se jedná o hmotné molekulární předpoklady. Tyto předpoklady jsou vlohy tedy geny. Znak určujeme nejen pomocí genu, ale rozhodující vliv utváří i vliv jeho prostředí. Při dědičnosti kvalitativních znaků bývá často tento znak závislý pouze na jednom genu.









Obrázek 2.5: Schéma dědičnosti konkrétního znaku (překresleno z [1])

2.1.4 Genetika psa

Má bakalářská práce se zabývá chovem, proto se nadále zaměříme spíše na dědičnost živočichů, konkrétně psa [2]. Pes je pátým savcem z 5500 druhů savců žijících na naší planetě, jehož genom byl rozluštěn v roce 2005. Pes má 20 439 genů, člověk má o 18% více. Geny psa tvoří 5% celé jeho DNA, zbytek DNA nic nekóduje. U psích plemen bylo popsáno 2,5 milionu polymorfismů, díky této variabilitě máme více než 400 druhů plemen psů (Plemena psů podle FCI – **Příloha 2**). Na druhou stranu způsobuje to i velké množství dědičně přenosných chorob a anomálií. Podle různých studií je u psů popsáno přibližně 500 dědičných chorob. Překvapivé je, že celá řada dědičných nemocí je podobná jako u člověka. Příčina mutace byla popsána zatím pouze u 50 nemocí z těchto chorob. Haploidní počet chromozomů psa je třicet devět, pokud vyjdeme z předpokladu, že je možná kombinace genů v chromozomu 2^{39} , můžeme mít více než 500 miliard kombinací.

Při šlechtění plemenných psů je nejdůležitějším předpokladem správná barva srsti. Některé barvy jsou nežádoucí i z hlediska zdraví psa, mohou značit vážnější nemoc. Někdy může mít barva srsti přímou souvislost se slepotou nebo hluchotou, také ale může barva srsti značit příměs příbuzného plemene. Barvu srsti a její zbarvení má na starosti 10 genů z 10 lokusů, jsou označeny: *A, B, C, D, E, G, M, P, S* a *T*. Pokud chceme zjistit dědičné zbarvení, můžeme se setkat se třemi problémy. Prvním z nich je, že chovatelé často používají jiný popis pro stejná zbarvení. Dalším problémem je neprokázaný původ, kdy se mohou objevit vlastnosti předka jiné rasy až u druhé či třetí generace. Třetím problémem je objasnění genetické podstaty zbarvení, toto často není možné bez cíleného křížení různých plemen, což je v čistokrevných rodech zakázáno.

Melanin s alelami tvoří základ zbarvení srsti. Přesné zbarvení zaleží na typu alely. Pokud se jedná o alelu *c*, bude jedinec albínem s bílou srstí, drápy, nepigmentovou kůží a červenýma očima. Jestliže má být pigmentace plná, je nutná přítomnost alely *C*. Dědičnost nejen zbarvení je velice komplikovaná a složitá, proto se ji pokusím znázornit pomocí **tabulky 2.2**. Dědičností zbarvení a různými kombinacemi se zabývá spousta odborné a specifikované literatury.

		Zlatý 	
		A^y	a^t
Černý 	A^s	 $A^s A^y$ černý	 $A^s a^t$ černý
	a^t	 $A^y a^t$ zlatý	 $a^t a^t$ s pálením

Tabulka 2.2: Očekávaný poměr potomstva 2:1:1 (převzato z [2])

Kromě barvy srsti a různých vlastností jsou děděny samozřejmě i choroby a různé dědičné defekty. Jedná se o všechny odchylky od typického fyziologického a anatomického stavu daného plemene. Tyto odchylky mohou mít menší či větší vliv na život. Psi se často používají při výzkumu dědičných chorob, protože se snadno aplikují i na člověka. Nežádoucí choroby a defekty mají vznik hned po splynutí pohlavních buněk, proto je důležité vědět, jak prochází daná nemoc či defekt celou populací plemene. Na toto se specializují chovatelské kluby, které mají nadřazenost nad chovem. Neexistuje plemeno, které by netrpělo některou dědičnou nemocí, proto je jejich výskyt sledován. Chovatelé se snaží tyto nemoci maximálně eliminovat, některé se objevují výjimečně a nejčastěji v rodových liniích. Pokud je nemoc ve vrhu podceněna nebo není zabráněno jejímu rozšíření, může to být závažný problém dalších generací. Podcenění některých chorob je velmi závažné, protože dochází k rychlému rozšíření po celé populaci. Následná náprava zabere více času a v nejhorších případech dokonce vede k zániku plemene. Mezi nejznámější a nejrozšířenější choroby patří dysplazie kyčelních kloubů, dědičná slepota, šeroslepost, hluchota, kýla, nadměrná plachost, předkus, podkus, nedostatečně vyvinutý chrup, progresivní atrofie sítnice, aj. Některé choroby se projeví až ve vyšším věku zvířete, ty choroby se dají těžce podchytit. Některé dědičné znaky jsou mnohdy uváděny jako plemenný znak určitého plemene. Takovýmto defektem může být například volné víčko u baseta a jiných plemen. To, že jde o defekt, dokazují časté infekce a záněty spojivky těchto plemen.

Je několik genetických aspektů, kterých je dobré sledovat při šlechtění plemen. Nejprve zjistíme, zda se jedná o nemoc dominantní či recesivní. Jedinci s dominantními znaky nemoci se z chovu jednoduše vyloučí, na rozdíl od jedinců s recesivními znaky. Jestliže jsou znaky recesivní znamená to, že jedinec je pouze přenašeč a nemoc se projeví až v následujících generacích. V takovýchto případech se pozoruje vždy celý vrh a rodiče se chovně omezují, nebo z chovu úplně vyřadí (**Tabulka 2.3**).

	+ / - x + / -		
	OBA KLINICKY ZDRAVÍ		
	OBA NOSITELÉ VLOHY		
POTOMSTVO	25% + / +	50% + / -	25% - / -
KLINICKY:	zdraví	zdraví	postižení
GENETICKY:	zdraví	nositelé vlohy	postižení

Tabulka 2.3: Schéma očekávaného potomstva s dědičnou nemocí (převzato z [2])

Pokud se v chovu projevuje více dědičných chorob, sestaví se postup, který určí, jaká onemocnění se odstraní nejdříve, aby byly následující generace méně postižené tou nemocí než jejich rodiče. Bohužel stačí jeden nositel vlohy a nemoc se může opět objevit u většiny štěňat. Mnohem

horší je situace, kdy je recesivní dědičná choroba kontrolována dvěma a více geny. Nápodobně je tomu, když se choroba projeví, pokud k tomu má vhodné prostředí. Komplikované je vymýcení genetické choroby, když je vázána na pohlaví.

2.1.5 Mutace

Pojmem mutace představuje strukturální a kvalitativní změnu genu. Dochází ke vzniku odlišného úseku DNA, tedy nové alely z původního genu. Mutace připadá na jednoho jedince z 10 000 až 100 000. Existují mutace příčinné, protože zapříčiňují nemoc u zvířete. Mutace letální a semiletální způsobují změny v životních orgánech a mohou zapříčinit smrt zvířete. Naopak některé mutace mají pozitivní vliv, pomáhají evoluci. Přispěly ke změnám a utváření různorodých plemen. Nonsense mutace nezpůsobují žádné změny, vytvářejí ale polymorfismus. Polymorfismu lze využít k ověření původu či identifikaci jedince. I mutace mohou být dominantní nebo recesivní a všechny mutace jsou dědičné.

2.2 Základy šlechtění plemen

Plemeno je populace zvířat stejného původu, které mají stejné či podobné vlastnosti, ty dědí jejich potomci a další generací [2]. Jedná se o populaci, která nepotřebuje příbuzenské křížení, aby byla schopna dále přežít. V této části se zaměřím především na šlechtění plemen psů. Nejen k šlechtění plemen psů se využívá znalostí z genetiky. Pomocí cílené plemenitby a vhodnou volbou rodičů lze dosáhnout zlepšení úrovně plemena. Pes byl podle mnoha předpokladů prvním domestikovaným zvířetem. Jak jsem se už zmínila v úvodu, úloha zvířat se stále více přizpůsobovala požadavkům člověka. Nejdříve psi pomáhali lovit, později se stali pomocníky při pastvě stád, také hlídali majetek a obydlí lidí, tím vznikala plemena lovecká, pastevecká a hlídací. Většina dnešních populárních plemen vznikla až v 19. a 20. století. Přibyla nová odvětví plemen, jakými jsou plemena společenská nebo dostihová. Nespočet chovatelských klubů bylo založeno koncem 19. století. Teprve až v minulém století byla sjednocena chovatelská pravidla čistokrevných zvířat. Došlo k registraci plemen a vytvoření plemenných knih. Začaly se konat velké výstavy a soutěže.

2.2.1 Selektce

Selektce je jeden z nejdůležitějších faktorů pro ovlivnění výsledků chovu. Působí na kvalitu dalších generací a celého plemene. Jedná se o výběr vhodných jedinců pro reprodukci. Jedinci, kteří jsou nežádoucí, mají nechtěné znaky či vlastnosti, jsou ze selektce vyřazeni. Selektční efekt je výsledek

selekce v budoucí generaci. Závisí na selekčním tlaku, dědičnosti selekčních znaků a vlastnostech plemene. Selekční tlak je rychlost selekce. Jedná se o hranice, které rozhodují, co je pro další reprodukci akceptovatelné. Pokud chceme provést selekci na dominantní znak, kdy tento znak označíme symbolem A , odstraníme z chovu jedince s AA a Aa , naopak jedinci aa mohou v reprodukci pokračovat, protože tento nežádoucí znak nepřenášejí. Recesivní znak se vyřazuje z chovu složitěji, je zde nutná řada vyšetření, protože někteří jedinci mohou být nositeli vlohy, čímž může dojít k vyřazení postiženého jedince, jeho rodičů i sourozenců. Tato selekce je časově náročná. Je více než důležité ujasnit si, co je pro chov prioritní, jaké vlastnosti chceme vždy dosáhnout. Může totiž nastat, že po vyřazení všech jedinců s nežádoucími vlastnostmi, nezůstane žádný jedinec pro další reprodukci.

Typy selekce jsou stabilizační, disruptivní a direkcionální. Stabilizační selekce se omezuje v reprodukci na jedince s extrémně vyvinutým znakem. Cílem je snížit rozmanitost plemene a dosažení vyrovnanosti vlastností psů a fen. Disruptivní selekce je opakem stabilizační. Snahou je vyšlechtit dva typy téhož plemene, např. dvě velikosti těla. Direkcionální selekce je mířena na určitý znak, např. u chrtů nejlepší vlohy pro běh a dosažení větší rychlosti.

Postupy selekce jsou tandémová, selekce podle nezávislých výběrových úrovní a simultánní selekce podle selekčních indexů. Snahou je dosáhnout rychlého úspěchu. Tandémovou selekcí se vybírají jedinci podle jednoho znaku, dokud tento znak není v celé populaci. Selekce podle nezávislých výběrových úrovní je postup, kde se stanovují limity v exteriéru, výkonu a zdraví. Jedinci, kteří těchto limit nedosáhnou, jsou z chovu vyřazeni. Používá se nejčastěji a je nejjednodušší. Simultánní selekce podle selekčních indexů udává číslo, které vyjadřuje výhodu nebo nevýhodu genotypu jedince. Je to způsob, jak odhadnout hodnotu jedince pro danou populaci. Vypočítat index je komplikované a složité, proto se občas využívá i odhadu.

Selekce je přesnější pokud jsou známy plemenné hodnoty předků, chovná a užitková hodnota jedince a jeho sourozenců a užitková a plemenná hodnota potomků jedince. Je mnoho selekčních metod a postupů, proto je důležitý individuální přístup, který si zvolí chovatel nebo klub.

2.2.2 Metody plemenitby

Mezi nejpoužívanější metody plemenitby patří čistokrevná plemenitba a křížení. Čistokrevná plemenitba je nejběžnější. Čistokrevný jedinec musí být zapsán v plemenné knize, která je schválena FCI, nebo American Kennel Club, British Kennel Club a Canadian Kennel Club. Musí mít také platný průkaz původu. Čistokrevná plemenitba se dále dělí na příbuzenskou a nepříbuzenskou plemenitbu. Příbuzenská plemenitba znamená, že v rodokmenu vyskytuje společný předek ze strany otce i ze strany matky. Existuje varianta příbuzenské plemenitby, tou je plemenitba liniová. Vychází z jednoho nebo více společných předků. Pokud jejich potomci mají žádoucí vlastnosti, jedná

se o liniovou plemenitbu. Má několik etap, první z nich je orientace v populaci, kdy se vytvoří genealogické linie. V druhé etapě se přechází na linii chovnou a vybere se hlavní znak nebo vlastnost. Samotné šlechtění prochází ve třetí etapě. O využití příbuzenské plemenitby jsou vedeny spory. Všeobecně platí, že ke šlechtění, obzvláště nových plemen, je příbuzenská plemenitba zásadní. Musí probíhat kontroly tohoto šlechtění, aby byl znám genotyp šlechtěných jedinců. Riziko výskytu nežádoucích znaků je při nedostatečném povědomí o genetických vlastnostech jedince nebezpečné. Existuje vzorec, kterým lze vypočítat stupně příbuznosti.

$$F_X = \sum \left[\left(\frac{1}{2} \right)^{n_1 + n_2 + 1} (1 + F_A) \right] \quad (1)$$

F_X ... koeficient příbuzenské plemenitby

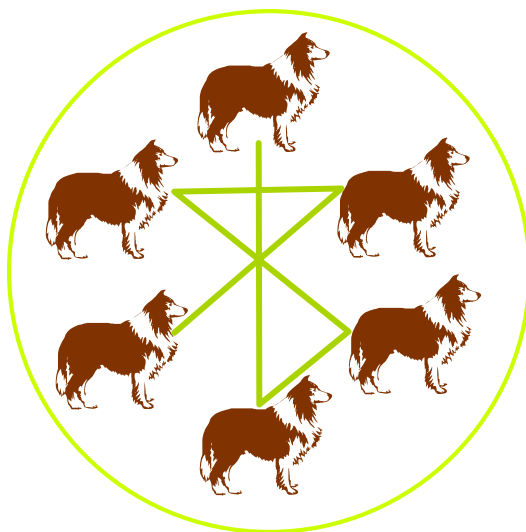
F_A ... hodnota F_X předka z příbuzenské plemenitby

n_1 ... počet pokolení společného předka na straně otce

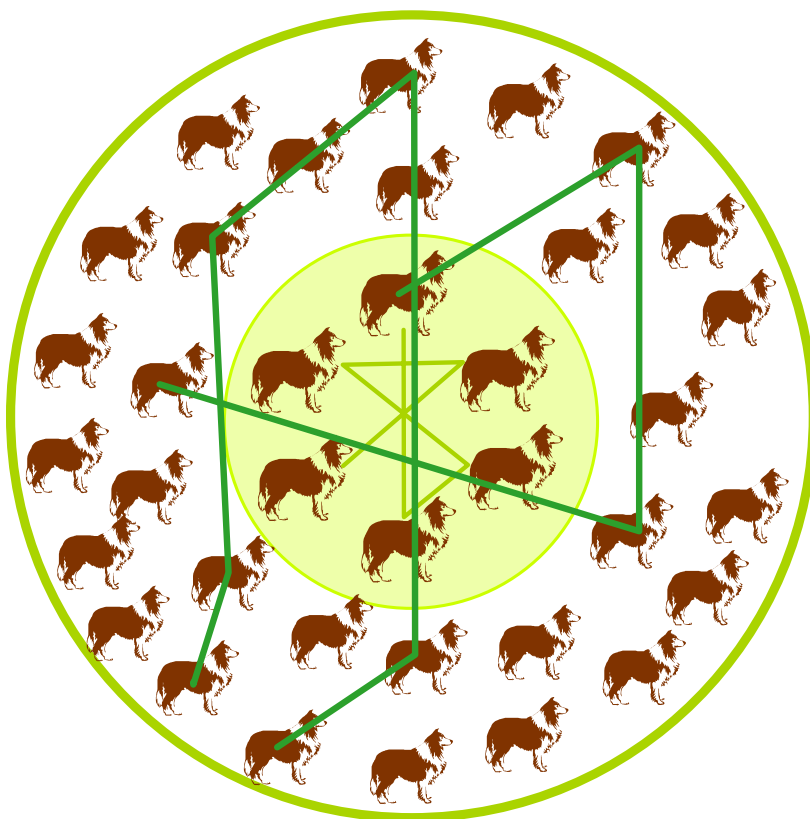
n_2 ... počet pokolení společného předka na straně matky

Křížení je páření dvou jedinců různých plemen. Nový jedinec může být zapsán do plemenné knihy, je označen za křížence stejně jako na průkazu původu. Křížení se používá k zušlechtění plemen, regenerace původních plemen, přilítí krve, pro užitkové a převodné křížení nebo vytvoření nového plemene. Regenerace původních plemen je obrácený postup než vznik nového plemene. Opět je nutné stanovit, čeho chce chovatel dosáhnout. Vyhledají se psy a feny, kteří mají požadované znaky. Tito jedinci se mezi sebou páří, až dojde k ustálení plemenného typu. Přilítí krve je povolováno FCI. Jedná se o přilítí krve původního plemene, kdy jsou jedinci páření jen jednou a další tři generace jsou vzájemně páření jen s jedinci původního plemene. Všichni tito jedinci jsou zapisováni do registru plemenné knihy. Čtvrtá generace může být pářena mezi sebou a je zapsána do řádné plemenné knihy. Užitkové křížení je zvláštní případ, kdy chovatel chce dosáhnout jedince s lepšími vlastnostmi, než které měli původní jedinci různých plemen. Převodné křížení je metoda plemenitby, kdy pro zušlechtění plemene jsou použiti jedinci jiného plemene. Výslední jedinci jsou nadprůměrní, proto jsou páření už jen s jedinci z druhého plemene. Toto křížení je zakázáno v zemích FCI. Každé nové plemeno, které chce být uznáno FCI, musí mít minimálně osm linií a v každé z nich aspoň šest fen a dva psi. Zároveň všichni jedinci nesmí být příbuzní do třetí generace předků. Existují tři etapy vývoje nového plemene. První etapou je vznik, kdy se vybírají původní plemena pro své vlastnosti a znaky. Tito jedinci jsou podle počtu páření úzkou nebo vzdálenější příbuzenskou plemenitbou. Tato etapa je znázorněna na **obrázku 2.6**. Druhou etapou je rozšíření, kdy je cílem dosáhnout vzdálené příbuzenské nebo nepříbuzenské plemenitby (**Obrázek 2.7**). Třetí etapou je zušlechtování. Provádí se selekce a jsou stanovena kritéria. Jednotlivé etapy se mohou vzájemně překrývat. Je důležité tedy vývojový proces stále sledovat, protože plemeno může při nedokonalé

a nedostatečné kontrole dědičných vlastností, znaků a nemocí zaniknout dříve, než bude uznáno za nové plemeno.



Obrázek 2.6: První etapa vzniku nového plemena (překresleno z [2])



Obrázek 2.7: Druhá etapa vzniku nového plemena (překresleno z [2])

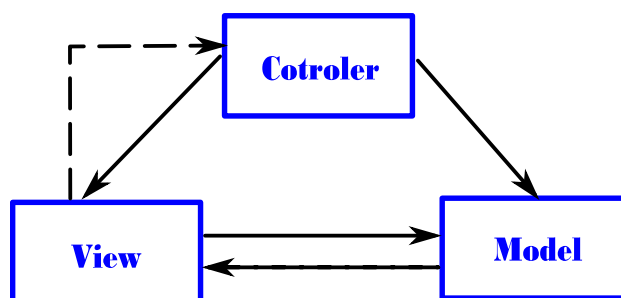
3 Technologie

V této kapitole popíši technologie a nástroje, které jsem použila. Informační systém je systém, který sbírá, udržuje, zpracovává a poskytuje informace a data uživatelům. Tento informační systém v podobě webové aplikace není nutné instalovat, není potřeba instalovat ani jiný speciální software. Je dostupný přes síť Internet pomocí webového prohlížeče. Webová aplikace je nejvhodnější, protože je rozšiřitelná a méně náročná pro pochopení uživateli. Je dostupná na většině operačních systémech. V této kapitole jsem čerpala ze zdrojů [8] až [21].

3.1 MVC model

V této práci využívám MVC modelu. MVC je zkratka Model – View – Controller [21]. Model znamená model informací, s těmito informacemi pracuje výsledná aplikace. View je pohled, je to pohled uživatele, kdy jsou mu předávány informace určitou formou. Pod pojmem Controller se označuje tzv. řadič. Controller manipuluje s Modelem, zpracovává data a pracuje s View. Často se využívá jako návrhový vzor pro webové aplikace, označuje se i za architektonický. Jeho implementace využívá frameworků.

MVC model (**Obrázek 3.1**) může pracovat různými způsoby, ale obecně platí následující postup. Uživatel provede nějaký výběr, stisk tlačítka. Z uživatelského prostředí dojde Controlleru oznámení, že došlo k akci. Ten přistoupí k modelu, pokud si to akce žádá, je model zaktualizován. Model zpracuje změnu a pošle informace View. View je na Modelu nezávislý, ale může se u něj zaregistrovat jako příjemce. Než se zobrazí konečné informace na prohlížeči, může dojít k žádosti o obnovení stránky. View čeká na další akci od uživatele, tím celý koloběh proběhne znovu. Různé frameworky spojují View a Controller různým způsobem, což vede k různým interpretacím, často i chybným.



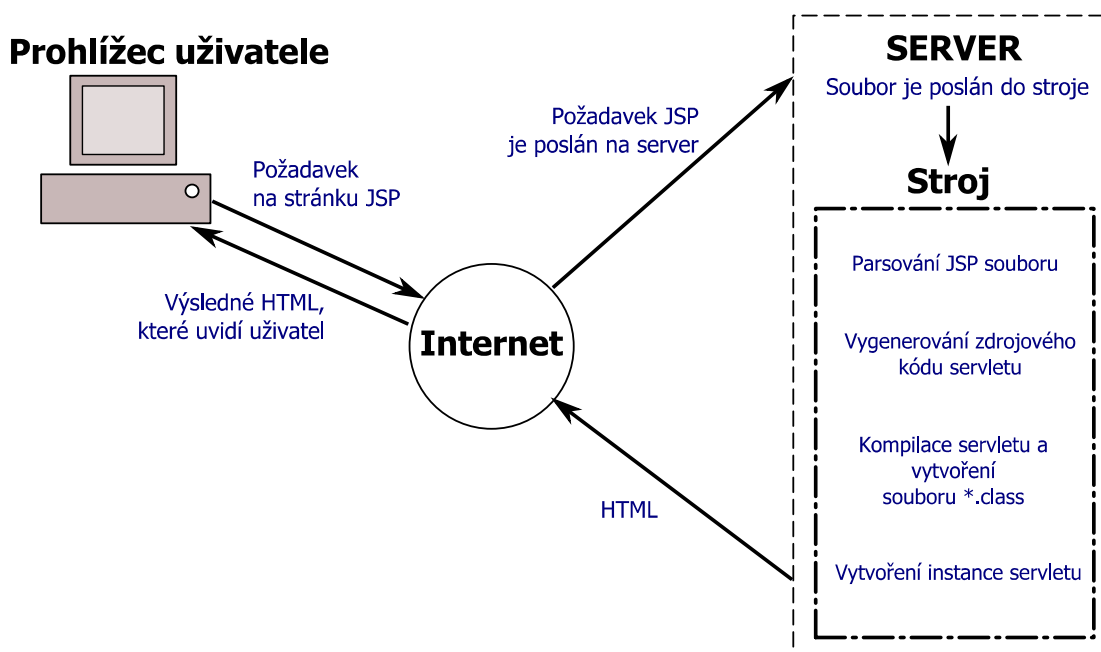
Obrázek 3.1: MVC model (překresleno z [21])

3.2 JavaServer pages

JSP byla vyvinuta společností Sun Microsystems [8], [16], [17]. Tvoří aplikace na straně serveru. Základem jsou HTML stránky. V těchto stránkách jsou obsaženy speciální tagy, které tvoří zdrojový kód v Javě. Díky těmto tagům jsou dynamicky vytvářeny stránky. JSP se upřednostňuje při statickém obsahu dokumentu.

Výhodou JSP je, že dynamická část je oddělena od statické. Výhoda oproti často používanému PHP je, že Java poskytuje rozsáhlé třídy na práci se sítí, s databázemi, elektronickou poštou aj. Předností oproti ASP je, že není potřeba učit se speciální jazyky VBScript atd. Java je oproti jazykům určených pro ASP komplexnější a bezpečnější.

Princip činnosti JSP je znázorněn na **obrázku 3.2**, kdy zdrojový kód se přeloží na servlet a zkompileje. Vytvoří se servlet a ten generuje HTML kód. Uživatel se dotáže a klient dotazem vytvoří požadavek – request. Ten je poslán na server. Server detekuje dotaz na JSP stránku a přesměruje JSP soubor do stroje JSP Servlet. JSP stroj zjistí, zda požadovaná stránka již byla přeložena. Pokud byla už přeložena, zavolá metody `init()` a `service()`. Pokud nebyla ještě přeložena, stroj stránku zkontroluje. Po zkontrolování je vygenerován servlet. Servlet je speciální a vytvořený podle JSP souboru. Obsah v HTML je převeden do příkazu `out.println()`. Po zkompileování servletu se vytvoří soubor “.class“. Vytvoří se instance servletu a jsou zavolány metody `init()` a `service()`. Uživateli se zobrazí HTML, které je výsledkem servletu.



Obrázek 3.2: Princip činnosti JSP(překresleno z [16])

3.3 Struts 2

Struts 2 je framework, který je založen na jazyce Java a slouží po vývoj webových aplikací [18]. Byl navržen pro kompilaci celého vývojového cyklu, včetně budování, rozvoje a udržování celé aplikace. Každá třída frameworku je založena na rozhraní. Všechny základní třídy mají dány speciální metodu, dokonce je možné přidat si vlastní. Rozdíly a jednotlivé vlastnosti Struts 1 a Struts 2 jsou popsány v tabulce 3.1.

Struts vznikl v roce 2000 za podpory Apache Software Foundation rozšířením klasických servletů pro MVC architekturu. Struts 2 vznikl v roce 2005 a nyní je vyvíjen pod licencí Apache Licence 2.0. Podrobnější informace a aktuální verze ke stažení jsou k nalezení na této stránce <http://struts.apache.org/2.1.8.1/index.html>.

VLASTNOSTI	STRUTS 1	STRUTS 2
Třídy action	<ul style="list-style-type: none"> - rozšiřuje abstrakci základních tříd na Action třídy - problémem je, že používá spíše abstraktní třídy, než rozhraní 	<ul style="list-style-type: none"> - Action třídy se implementují pomocí Action rozhraní - rozhraními dovolují použít optimální a uživatelské služby - použití Action rozhraní není nutné
Model vláken	<ul style="list-style-type: none"> - Action jsou typu "singleton", proto musí být zajištěna bezpečnost (z hlediska vícenásobného přístupu) vlákna, protože pouze jedna instance třídy ovládá všechny požadavky pro Action - nutná zvýšená obezřetnost vůči bezpečnosti, synchronizace a rozvoje aplikace 	<ul style="list-style-type: none"> - neexistuje problémy se zabezpečením vláken (z hlediska vícenásobného přístupu), protože pro každou Action třídu se vytvoří instance pro každý požadavek - Servlet container vytváří mnoho objektů, které se po requestu zahodí, jeden objekt navíc neznamena snížení výkonu nebo dopad na "garbage collection"
Závislost Servletu	<ul style="list-style-type: none"> - Action jsou závislé na API, protože HttpServletRequest a HttpServletResponse jsou předány metodě execute, když se vyvolá Action 	<ul style="list-style-type: none"> - Servlety kontextů jsou obvykle reprezentovány jako jednoduché mapy, které umožňují Action testovat izolovaně - Action mohou stále využívat původního požadavku a odpovědi v případě potřeby - ostatní architektonické prvky přímo snižují nebo eliminují potřebu přístupu k HttpServletRequest nebo HttpServletResponse.
Testovatelnost	<ul style="list-style-type: none"> - hlavní problém, protože při testování aplikace je metodou execute odhaleno Servlet API - Struts TestCase poskytuje sadu falešných objektů 	<ul style="list-style-type: none"> - při testování Action se vytvoří instance Action, nastaví vlastnosti a zavolají se metody

Vstupy	<ul style="list-style-type: none"> - obdrží vstup vytvořený objektem ActionForm - nejen všechny třídy ActionForm musí být rozšířeny o základy třídu ActionForm - ostatní třídy JavaBeans třídy nemohou být použity jako ActionForm, zatímco vytváříme redundantní třídy k získání vstupu - DynaBeans je nejlepší alternativou k vytvoření tradiční ActionForm třídy. 	<ul style="list-style-type: none"> - vyžaduje vlastnosti Action jako vstupní vlastností, tím že eliminuje potřebu druhého vstupu - tyto vstupní vlastnosti mohou dosáhnout objektových typů, protože mohou mít vlastní vlastnosti - můžeme přistoupit k vlastnostem Action z webové aplikace pomocí tagů - podporuje také ActionForm, formát POJO Actions aj.
Výrazový jazyk	<ul style="list-style-type: none"> - obsahuje JSTL, takže používá JSTL EL - základní objekt má v EL graf průchodnosti, ale poměrně slabou kolekci a podporu indexovaných vlastností 	<ul style="list-style-type: none"> - může využít JSTL, ale rovněž podporuje silnější a flexibilnější jazyky, nazývané "Object Graph Notation Language" (OGNL)
Přiřazení hodnot do view	<ul style="list-style-type: none"> - Struts 1 váže objekty do kontextu stránek pomocí standardních JSP mechanismů 	<ul style="list-style-type: none"> - Struts2 používá ValueStack, aby hodnoty byly přístupné Tagům bez vazby na objekt, který je vytváří - strategie ValueStack umožňuje znovu použít views napříč typy, tím že existuje stejná proměnná, ale různých typů
Typová konverze	<ul style="list-style-type: none"> - vlastnosti ActionForm jsou ve formě řetězců - pro typovou konverzi jsou používány Common – Beanutils 	<ul style="list-style-type: none"> - používá OGNL pro typovou konverzi a konvertory ke změně základních a veřejných objektových typů
Validace	<ul style="list-style-type: none"> - Struts 1 používá ruční ověřování, které se provádí pomocí validace metodou na ActionForm, nebo pomocí rozšíření Commons Validator. - třídy mohou mít různé validační kontexty pro stejné třídy, přičemž zřetězení k validaci sub-objektů není dovoleno 	<ul style="list-style-type: none"> - Struts 2 umožňuje manuální validace, která se provádí pomocí validační metody a frameworku XWork Validation - framework XWork Validation umožňuje zřetězení validací do sub-objektů pomocí validací definovaných pro vlastnosti typu tříd a validace kontextu
Kontrola Action výjimek	<ul style="list-style-type: none"> - každý modul v Struts 1 má zvláštní procesory žádost (životní cyklus), zatímco všechny akce v modulu musí sdílet stejný životní cyklus 	<ul style="list-style-type: none"> - ve Struts jsou 2 různé životní cykly vytvořené na základě Action pomocí Interceptoru - vlastní tunely jsou vytvářeny a používány pro různé akce, jak se vyžaduje

Tabulka 3.1: Rozdíly mezi Struts 1 a Struts 2 (převzato z [19])

Spring na rozdíl od Struts 2 nebyl vytvořen pouze pro tvorbu webových aplikací [20]. Spring je spíše integrační framework. V současné době je vyvíjen společností SpringSource pod licencí Apache License 2.0. Pro Spring existuje spousta rozšíření, která usnadňují práci na konkrétním typu aplikace. Jádru Springu je použitelné v každé aplikaci.

3.4 Ajax

AJAX se využívá pro vývoj moderních webových aplikací, kdy využívá Javascript pro zobrazení dynamických změn [9]. Pro přenos informací z klienta na server, případně opačně, využívá XML soubory. Díky tomu je možné odeslat data na pozadí aplikace a přijmout data ze serveru bez potřeby načtení celé stránky. AJAX také přináší jisté nevýhody. Hlavní nevýhodou je problém s tlačítkem Zpět, které nepracuje tak, jak by uživatel očekával. Je nutné řešit to speciálně, což přináší další práci navíc při implementaci. Dalším problémem je neměnicí se URL adresa při přechodu mezi stránkami. Uživatel nemůže poslat odkaz na stránku, např. emailem.

Při práci na komplexní aplikaci by byla klasická implementace pomocí AJAXu příliš časově náročná, proto se často využívá různých frameworků. Struts 2 nabízí podporu AJAXu pomocí frameworku DOJO [22]. Tento framework poskytuje mnoho různých pomůcek pro práci s AJAXem, např. autocompleter nebo taby. To programátorovi velmi ulehčí práci, protože se nemusí zabývat implementačními detaily.

3.5 HTML, CSS, Javascript

HTML (Hypertext Markup Language) je hypertextový značkovací jazyk, patří do rodiny jazyků SGML(Standard Generalized Markup Language) [10]. SGML je metajazykem pro značkovací jazyky. Značkovací jazyky jsou abstraktní nástroje pro vložení informací do běžného textu. První verze HTML vznikla současně s architekturou WWW. HTML představuje standard tvorby hypertextových dokumentů a základní nástroj pro tvorbu webových aplikací. HTML se skládá z tagů a atributů. Tagy jsou značky, které označují a formátují text, mohou být párové nebo nepárové. Atributy jsou vlastnosti těchto značek. Prostředky pro formátování textu jsou řádkové a blokové elementy.

S rozšířením Internetu a stále větším využíváním jazyka HTML se objevily problémy, HTML se stal nepřehledný a komplikovaný. Struktura dokumentu nebyla oddělena od vizuálních prvků. Kaskádové styly (Cascading Style Sheets - CSS) tyto problémy odstranily, ale nejsou podporovány ve všech starších verzích prohlížečů. CSS jsou nástrojem, který upřesňuje vzhled vybraných HTML

elementů v dokumentu. Základem je stylový předpis. Je to textový soubor s posloupností pravidel. Tyto pravidla určují vzhled různých elementů.

JavaScript je skriptovací jazyk [12], [13]. S jazykem Java má jen společného předka C/C++, jinak se jedná o různé jazyky. Je to jazyk interpretovaný, nemusí být tedy kompilován a program je vykonán přímo prohlížečem. Podporuje objektově orientované programování. Nezatěžuje server, protože pracuje na straně klienta. Používá se ke zvýšení interaktivnosti webových stránek, kontroluje formuláře před odesláním na server, spouští různé programy a ovládá webové prohlížeče na reakci uživatele, vytváří dynamické stránky, aj. Může se zapsat přímo do zdrojového souboru nebo do souboru s koncovkou .js. Nemůže pracovat se soubory na disku kvůli bezpečnosti. Uživatel může kdykoliv Javascript na svém počítači vypnout.

4 Návrh aplikace

V této kapitole je popsána analýza a návrh informačního systému pro evidenci chovu. Jsou zde upřesněny požadavky na systém. Také zde bude uveden use case diagram, který zobrazuje pohled na systém. V této kapitole jsem čerpala ze zdrojů [8] až [21].

4.1 Existující systémy pro evidenci chovu

Existuje mnoho webových portálů, které sdružují chovatele, málokteré ale umožní chovatelům vytvářet rodokmeny. Tuto službu poskytují různé programy, které jsou ve většině případů placené. Největší nadnárodní firmou, která nabízí různé systémy pro zemědělce, je GEA Farm Technologies (<http://www.westfalia.com/cz/cz/>). Tato firma poskytuje systémová řešení a servis pro úspěch klientů. Vytváří systémy, které hlídají kvalitu mléka, užítkovost stáda, také srovnávají výrobní a provozní náklady. Problémem je, že se spíše specializují na dojení techniku pro krávy, ovce, kozy, aj. To je zcela nevhodné pro chovatele drobného zvířectva. Chov teraristických zvířat pomáhá organizovat například Darlinx (<http://www.instaluj.cz/darlinx-4-free>). Tato aplikace ukládá informace o chovu, ale pomáhá chovatelům i s dobou a množstvím krmení. Přesto postačuje pouze pro potřeby malého chovu. Existují i pomocníci pro chovatele holubů ve formě MS Excelu [7], kdy je pomocí tabulek a maker vytvořen systém, který dovoluje vytvářet i rodokmeny. Dalším programem je PC FAND (<http://www.alis.cz/fand/index.jsp>), který byl vytvořen pro uživatelské prostředí DOS. Aplikace je variabilní a lze ji kdykoliv uživatelsky přizpůsobit, potřebuje menší čas na zaškolení. Uživatel si zde vytváří vlastní databázový systém.

4.2 Požadavky na systém

Tento systém musí být přístupný více skupinám uživatelů. Webová aplikace je v tomto případě nejlepší volbou, protože Internet je v dnešní době značně rozšířen. Webovou aplikaci není nutné instalovat, což je dalším přínosem. Základním úkolem této aplikace je pomoci chovatelům uspořádat svoje chovná zvířata, případně dokonce vytvořit rodokmen jedinců. Jednotliví uživatelé mohou navazovat mezi sebou přátelství, mohou se stát členy různých skupin a mohou založit svoje vlastní skupiny.

V první řadě jsem se zaměřila na vzhled vytvářené aplikace. Aplikace musí být přístupná a uživatelsky příjemná. Tématem jsou zvířata, proto jsem zvolila jako hlavní barvu světle zelenou. Neocenitelným pomocníkem mi byla neplacená stránka Color Scheme Designer (<http://www.colorschemedesigner.com/>), kde jsou návrháři nabídnuty různé barevné variace.

Uživatelé jsou nabídnuty nejvhodnější barvy k původním zvoleným, které se dají sladit do různých barevných tónů. Také je zde nabídnuta přímo ukázková stránka s použitím zvolených barev. Podle této stránky jsem zvolila světlejší a tmavší odstíny a zároveň kontrastnější červenou. Celá stránka je laděna do zelených odstínů. Na pozadí jsou přidány prvky, které by uživatelům měly připomínat louku.

Dalším krokem bylo vytvoření loga (**Obrázek 4.1**). V dnešní době je nutné se zviditelnit, proto musíme potenciální uživatele nejdříve oslovit. Logo jsem vytvořila sama, zvolila jsem pochopitelně motiv zvířat. Základními předpokládanými uživateli byli chovatelé psů a holubů, proto základní motiv tvoří siluety holuba a psa. Tato aplikace předpokládá využití i chovateli zemědělských zvířat, proto je v pozadí ztvárněn tur domácí. Barva je světle zelená s kontrastním červeným názvem stránky. Aplikace je navržena tak, aby její ovládání bylo uživateli příjemné a zároveň jednoduché. Po přihlášení se načte úvodní stránka, na této stránce se potom pouze mění obsah, což pomáhá k rychlejšímu načítání stránek.

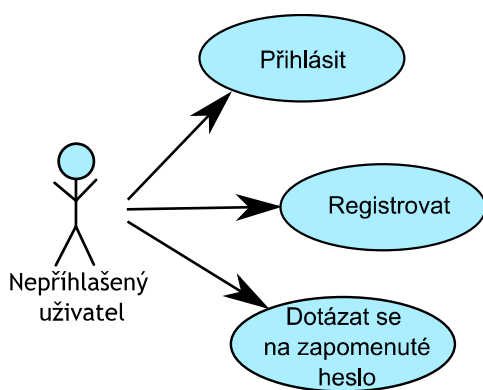


Obrázek 4.1: Logo

Vytvořit evidenci plemen různých živočišných druhů je velmi komplikované. Každý druh má vlastní rozdělení. Hospodářská zvířata, především tur domácí, lze rozdělit podle stupně šlechtění, podle směru užitkovosti, podle původu a podle teritoriálního rozšíření. Máme tedy skot primitivní, zušlechtěný a kulturní podle stupně šlechtění, skot stepní, nížinný, horský podle teritoriálního zaměření a plemena mléčná, masná a kombinovaná podle užitkovosti. Podle původu se rozděluje skot na pratuři, krátkorohé, krátkohlavé, čelnaté, dlouhočelnaté a bezrohé. Nejčastěji hospodářská zvířata rozdělujeme podle užitkovosti. U všech plemen, nejen hospodářských, evidujeme druh, pohlaví, zbarvení a rodiče, případně jiné jedinečné vlastnosti. U šlechtěných plemen psů by dokonce měl být znám rodokmen. U poštovních holubů kromě předků se ukládají i informace o nalétaném počtu kilometrů. Kvůli těmto různým vlastnostem využívám v systému vlastnost jedince. První vlastnost je nastavena jako prioritní a při vyhledávání se systém zaměřuje na ni. Také je důležité uložit uživatelům různá práva. Někteří uživatelé budou mocí pomocí těchto práv spravovat skupiny. A to ať už omezené množství informací nebo vše, co se skupinami souvisí.

4.3 Use case diagram

Use case diagram (model případů užití) se používá pro zachycení požadavků, popisuje funkcionalitu systému. V tomto případě se specifikují dva případy užití. První model je pro nepřihlášeného uživatele, tento model je znázorněn na **obrázku 4.2**. Nepřihlášený uživatel nemá přístup do systému, proto se musí nejdříve přihlásit. Před prvním přihlášením je nutná registrace. Pokud uživatel zapomene heslo, může požádat o zaslání nového hesla na email, email je současně přihlašovacím jménem.

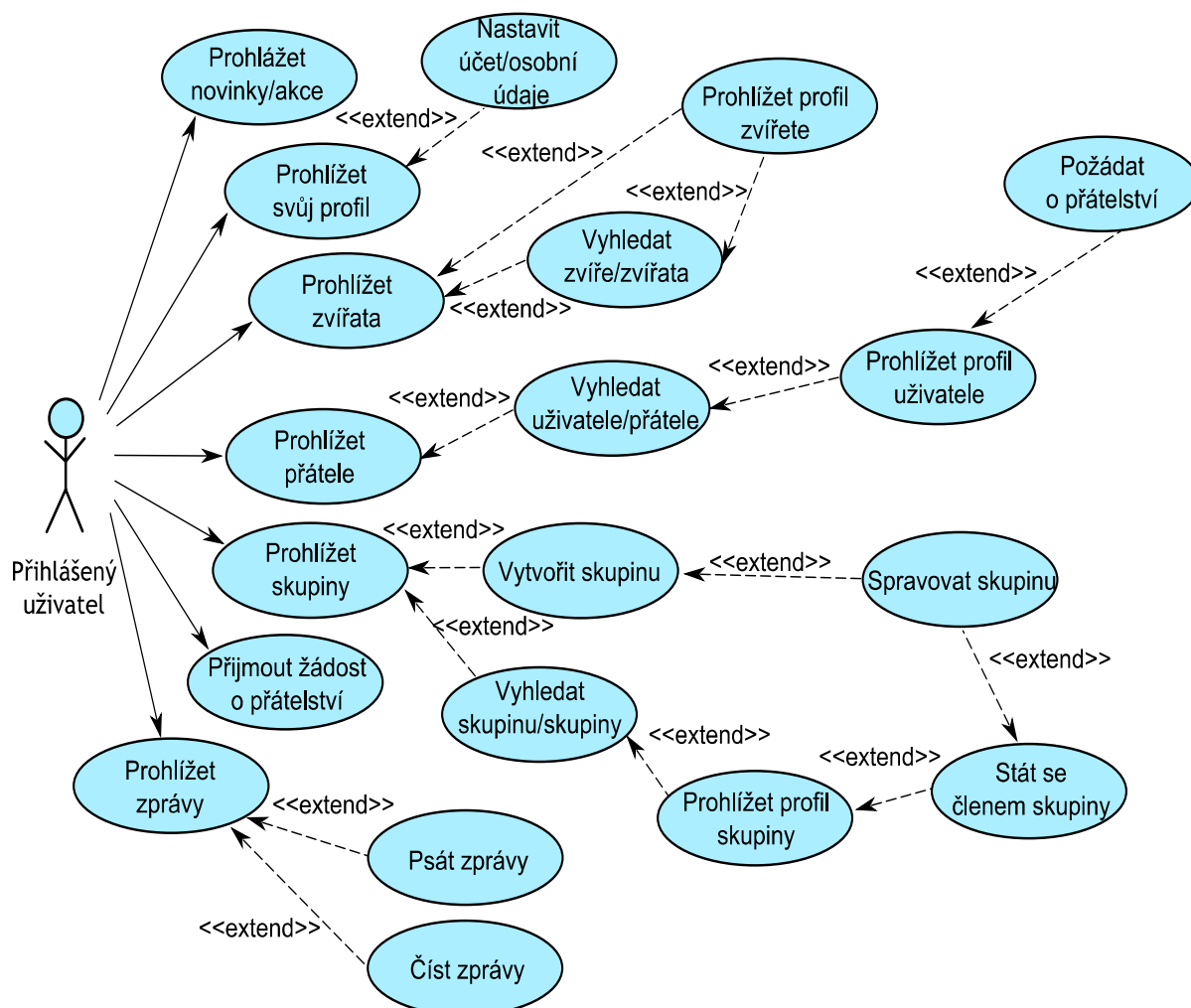


Obrázek 4.2: Use case diagram nepřihlášeného uživatele

Další use case diagram ukazuje možnosti přihlášeného uživatele (**Obrázek 4.3**). Případy užití pro přihlášeného uživatele jsou následující:

- **Prohlízet novinky/akce** – po přihlášení se uživateli automaticky zobrazí novinky, které se staly od doby posledního přihlášení, později si může všechny novinky/akce opět vyvolat
- **Prohlízet profil uživatele** – uživatel si může prohlédnout svůj profil, případně si ho podle potřeb a přání nastavit či změnit
- **Prohlízet zvířata uživatele** – v tomto případě jsou zobrazena uživateli jeho vlastní zvířata, má zde také možnost vyhledávat a to jak z vlastních zvířat, tak i ze zvířat přátel i cizích uživatelů. Může si prohlédnout profil jednotlivých vlastních zvířat nebo po vyhledání i nalezených jedinců.
- **Prohlízet přátele** – zde je uživateli zobrazen seznam jeho přátel. I v tomto případě má možnost uživatel vyhledat osoby, které jsou registrovány v systému. Uživatel může prohlízet profily dalších osob. Jestli-že osoba prohlíženého profilu není přidána mezi přátele přihlášeného uživatele, může tento uživatel zažádat o přátelství.
- **Prohlízet skupiny** – v tomto případě jsou vypsány uživateli skupiny, kterých je členem. Opět zde uživatel může vyhledat ze všech skupin, jichž je nebo není členem. Může si prohlédnout stránky skupiny a případně zažádat o členství.

- **Přijmout žádost o přátelství** – uživatel je upozorněn, že by si ho někdo rád přidal mezi přátele. Poté se uživatel může rozhodnout, zda přijme nebo odmítne
- **Prohlížet zprávy** – uživatel si může prohlížet nepřečtené a přečtené zprávy, nebo napsat zprávu novou. Může poslat zprávu jak přátelům tak cizím uživatelům



Obrázek 4:3: Use case diagram přihlášeného uživatele

4.4 ER diagram

ER diagram nebyl čistě součástí mé práce a vychází ze spolupráce s Radim Rešem [24] pod vedením Ing. Petra Chmelaře. ER- diagram, podle kterého pracuje moje aplikace, se nachází v **Příloze 3**.

5 Implementace

V této kapitole je popsána implementace aplikace. Je zde rozebrána aplikace a její části. Bude zde popsána komunikace mezi jednotlivými částmi modelu MVC, hlavně mezi View a Controllerem. Nakonec popíši problémy při tvorbě aplikace. Aplikace je vyvíjena pomocí Java 2 Enterprise Edition (J2EE). Je také použit framework Struts 2 a AJAX. Aplikace byla vytvořena ve vývojovém prostředí NetBeans 6.8, které je vyvíjeno společností Oracle (<http://netbeans.org>).

Nejdříve je popsána část Controller, která má na starosti komunikaci mezi databází a stránkami systému, které vidí uživatel. Ve Struts 2 Controller přijímá akce vytvářené uživatelem, podle toho řídí Model. Každý http požadavek je směřován do Controlleru, který rozhodne, jakou Action vyvolá. Action může mít přiřazenou Action třídu, potom při vyvolání Action je vytvořena instance Action třídy. Implementace databáze vychází ze spolupráce Radima Reše [24]. Úvodní strana mé aplikace je na **obrázku 5.1**.



Obrázek 5.1: Úvodní stana

5.1 Session

Session udržuje kontext při práci uživatele s webovými stránkami. Při přihlášení uživatele je vytvořen objekt Chovatel, který je poté umístěn do Session. Objekt Chovatel uchovává informace o právě přihlášeném uživateli, hlavně jeho email a jméno. Tento objekt je pak přístupný z View i z Action tříd. Pomocí toho se může zobrazit na stránce jméno přihlášeného uživatele a může se identifikovat uživatel při komunikaci s databází. Je nutné provádět testování existence objektu chovatel v Session, aby nepřihlášený uživatel byl vždy přesměrován na stránku Přihlášení, pokud se objekt Chovatel

v Session nenachází. V případě neaktivnosti uživatele je tímto automaticky odhlášen. Tento test je prováděn vždy na začátku metody `execute()`. Při odhlášení se objekt Chovatel pouze odstraní ze Session.

5.2 Profil uživatele

Pokud si uživatel chce prohlédnout svůj profil (**Obrázek 5.2**), je potřeba načíst informace o jeho datech z databáze. Je potřeba vytvořit instanci třídy `Mchovatel` a použít její metodu `getChovatel()`, která vrátí objekt `Chovatel` s naplněnými daty. Zvlášť se musí načíst jeho adresy a kontakty pomocí metod `get()` třídy `Madresa` a metody `get()` třídy `Mkontakt`. Nyní je vše připraveno na zobrazení profilu uživatele.

<u>Email:</u>	q.sixxao@seznam.cz			
<u>Jméno:</u>	Radim Reš			
<u>Počet zvířat:</u>	4			
<u>Počet skupin:</u>	124			
	<u>Ulice</u>	<u>Město</u>	<u>PSČ</u>	<u>Země</u>
<u>Adresa: 1</u>	Zábřežská 34	Šumperk	78701	420
<u>Adresa: 2</u>	Zábřežská 12	Šumperk	78701	420
<u>Adresa: 3</u>	Horní 9	Šumperk	78701	420
<u>Adresa: 4</u>	Prázdná 7	Šumperk	78701	420
<u>Kontakt</u>				
<u>Telefon</u>	777888999			
<u>WWW</u>	www.seznam.cz			
<u>Změnit</u>				

Obrázek 5.2: Detail profilu

Uživatel si může svůj profil změnit, je tedy nutné mu nabídnout upravitelné pole s již vyplněnými daty, které může uživatel změnit. Může si zde vytvořit novou adresu nebo kontakt. Uživatel může mít více adres a kdykoliv si může do předpřipraveného prázdného pole přidat další. Toto pole je vždy kontrolováno, zda je vyplněno. Pokud je vyplněno je nová adresa uložena do databáze pomocí metody `set()` třídy `Madresa`.

5.3 Zvířata

Pro zobrazení základních informací o zvířatech vlastněných přihlášeným uživatelem, je potřeba načíst všechny jedince z databáze pomocí metody `get()` třídy `Mjedinec`. Tato metoda vrátí naplněný seznam všech vlastněných zvířat uživatelem. Tento seznam potom lze jednoduše ve View procházet pomocí iterátorů a vypisovat všechny zvířata (**Obrázek 5.3**). Zde je možné kliknutím na odkaz prohlednout detail zvoleného jedince. Tento odkaz vyvolá Action třídu `AzvireDetail`, která načte a zobrazí detailní popis zvířete (**Obrázek 5.4**). Třídě `AzvireDetail` je předán parametr `sid` zvířete, které má být zobrazeno. Tento parametr potom v Action třídě je získán pomocí `ServletContext.getParameter(id)`. Obdobně se provádí smazání zvířete, kde se zavolá Action `AsmazZvire`. I zde je nutné zavolat metodu `del()` třídy `Mjedinec`. Je možné si vytvořit nové zvíře. Při jeho vytváření jsou uživateli nabídnuta plemena pomocí autocompleteru. Autocompleter byl vytvořen pomocí tagu `<s:autocompleter>`, kterému je potřeba předat jako parametr seznam existujících plemen. Uživatel může vložit obrázek zvířete. Tento obrázek je poté potřeba zmenšit na přednastavenou velikost, aby se celý vešel do výřezu zobrazení.



<u>Jméno zvířete:</u>	<u>Datum narození:</u>			
Ron	2.2.1999			
Flik	2.2.1999			
Jack	2.4.2010			
Red	12.9.2009			
Julia	4.4.2009			

Obrázek 5.3: Detail seznamu zvířat

<u>Foto:</u>	
<u>Popis:</u>	jménuje se Václav
<u>Vlastnosti:</u>	je hodný
<u>Zařazení:</u>	Pes: hovawart
<u>Narození:</u>	25.4.2000
<u>Umrті:</u>	
<u>Vlastní od:</u>	25.4.2001
<u>Vlastní do:</u>	
<u>Otec:</u>	Neznámý
<u>Matka:</u>	Neznámá
<u>Pohlaví:</u>	Samec
<u>Sex:</u>	

[Zpět](#)

Obrázek 5.4: Detail zvířete

5.4 Zprávy

Uživatel může vytvořit zprávu a poslat ji jiným uživatelům. Nejprve se mu v nabídce zobrazí všechny přijaté zprávy (**Obrázek 5.5**), aby si mohl procházet odeslané zprávy, musí přejít na položku *Odeslané*. Seznam příchozích zpráv se načítá z databáze pomocí metody `getZpravy()` třídy `Mzprava`. Pro odeslané zprávy je použita metoda `getOdeslane()`. Pokud přijde nová zpráva, zůstává označená jako nepřečtená, dokud uživatel nezvolí *Detail zprávy*. Nepřečtená zpráva se označí jako přečtená pomocí metody `precteno()` třídy `Mzprava`. Při vytvoření nové zprávy uživatel musí zadat email uživatele, kterému má zpráva dojít (**Obrázek 5.6**). Pro usnadnění je uživateli nabídnut seznam jeho přátel pomocí autocompleteru. Zpráva se odešle pomocí metody `create()` třídy `Mzprava`.

Nova zprava **Odeslane**

<u>Předmět:</u>	ahoj
<u>Od:</u>	q.sixxao@seznam.cz
<u>Obsah:</u>	Jak se pořád máš? A co Bora, už má zdra
<u>Prečteno:</u>	NE
<u>Datum:</u>	18.4.2010 14:59
Detail	
<u>Předmět:</u>	ahoj
<u>Od:</u>	siao@seznam.cz
<u>Obsah:</u>	Jak se máš?
<u>Prečteno:</u>	ANO
<u>Datum:</u>	17.4.2010 17:45
Detail	

Obrázek 5.5: Příchozí zprávy

<u>Komu:</u>	q.sixxao@seznam.cz ▼
<u>Předmět:</u>	ahoj
<u>Obsah:</u>	<p>Jak se pořád máš?</p> <p>A co Bora, už má zdravou tu nohu?</p> <p>Karolína už vrhla štěněta, máme tři psíky a dvě fenky. Jsou zdraví a ryychle rostou.</p> <p>Měj se dobře a někdy se ozvi</p> <p>Zdraví Radim</p>
<input type="button" value="Odeslat"/>	

[Zpět](#)

Obrázek 5.6: Psaní nové zprávy

5.5 Přátelé

Uživatelé mohou vytvářet přátelství. Uživateli je zobrazen seznam jeho přátel (**Obrázek 5.7**), načtení seznamu má na starosti metoda `getChovatele()` třídy `Mpratelstvi`. Pokud si chce uživatel přidat nového přítele, je nucen ho nejdříve vyhledat ze všech uživatelů této aplikace. To probíhá pomocí metody `find()` třídy `Mchovatel`. Této metodě je předán vyhledávaný řetězec, metoda vrátí seznam všech chovatelů, jejichž jméno nebo email obsahují hledaný podřetězec. U nalezených uživatelů je zobrazeno jméno, město a email. Uživatel si může vybranou osobu hned přidat nebo se podívat na jeho detail a přidat si ho potom z profilu. Požadavek na přátelství je vytvořen pomocí metody `create()` třídy `Mpratelstvi`. Počet žádostí o přátelství přihlášeného uživatele jsou zobrazeny v pravém horním rohu pod vyhledáváním, tato informace se obnovuje každých pět minut. Uživatel může na tento odkaz kliknout a zobrazí se mu seznam cizích uživatelů, kteří ho o přátelství žádají. Toto má na starosti Action `AprateleZadosti`, která volá metodu `getZadatele()` třídy `Mpratelstvi`. Uživatel může žádost přijmout nebo odmítnout. O odmítnutí přátelství se stará metoda `zamitnout()` třídy `Mpratelstvi`. Žádost je přijata pomocí metody `potvrdit()` třídy `Mpratelstvi`. Při přijmutí či odmítnutí žádosti je potřeba ihned aktualizovat počet žádostí. Toho se docílí tím, že tag `<s : div>` naslouchá na událost „topic“. Kliknutí na přijmout nebo odmítnout tuto událost vyvolá.

Vyhledat nové přátele:	
<input type="text"/>	
<input type="button" value="Vyhledat"/>	
<u>Email:</u>	q.sixxao@seznam.cz
<u>Jméno:</u>	Radim Reš
<u>Město:</u>	Šumperk
Detail	
<u>Email:</u>	R.entl@centrum.cz
<u>Jméno:</u>	Radka Entlová
<u>Město:</u>	
Detail	

Obrázek 5.7: Detail seznamu přátel

5.6 AJAX

Podstatnou část implementace tvoří práce s AJAXem. AJAX je implementován pomocí DOJO toolkitu, který je součástí Struts2. Celé menu se nemění, mění se pouze obsah centrálního prvku, který je představován jedním tagem `<div>`. Pomocí tohoto tagu není zapotřebí načítat stále prvky na stránce znovu. Tím se zrychlí načítání stránek. Odkazy jsou řešeny pomocí tagu `<s : a>`, kterému je potřeba nastavit atribut `theme = "ajax"`. Do atributu `href` je potřeba vložit odkaz na tag `<s : url>`, který odkazuje na Action nebo na JSP stránku. Pomocí atributů `loadingText` a `errorText` můžeme nastavit zprávu, jenž se zobrazí uživateli při načítání dat, respektive při chybě. Obsah vrácený pomocí odkazu tagu `<s : a>` je vložen do `<div>`, jehož `id` je potřeba přiřadit atributu `targets`. Obsah je potom zobrazen uživateli. Kromě odkazu je na pozadí potřeba také odesílat data zadaná pomocí formuláře. Toto se provede záměnou klasického tagu `<submit>` za `<s : submit>`, jehož `theme` parametr je nastaven na AJAX.

5.7 View

Díky modelu MVC se ve View nevyskytuje téměř žádný kód napsaný v Javě. Vše je ošetřeno pomocí Struts 2 tagů. Nejčastěji používaným tagem byl `<s : iterator>`. Tomuto tagu jako atribut `value` přiřadíme jméno seznamu, který jsme načetli v Action třídě. Tag přes tento seznam iteruje a načtenou položku vloží na `valueStack`, odkud je potom jednoduché hodnotu získat. Dalším používaným tagem byl `<s : if>`, jehož atribut `test` obsahuje výraz. Pokud je výraz pravdivý, je tělo tagu `<s : if>` zobrazeno uživateli na stránce.

5.8 Problémy při tvorbě aplikace

Při vytváření nového zvířete se zasílá obrázek zvířete na server. Zasílání je potřeba provést na pozadí prostřednictvím AJAXu. Klasická implementace s použitím Struts 2 a DOJO nefungovala. Na stránce se zobrazí pouze „[object HTMLDocument]“ a soubor se na server nenačte. Je to způsobeno chybou v originální implementaci Struts 2. Řešení tohoto problému bylo popsáno na fóru <http://ReadList.com>, kde byla funkce JavaScriptu řešící tento problém. Tuto funkci jsem použila při implementaci. Dalším problémem bylo tlačítko Zpět, které nefunguje standardním způsobem. Dalším složitějším problémem byla neměnitelnost URL adresy.

6 Testování

Je důležité otestovat systém během vývoje. Mohou se tím odhalit menší i větší nedostatky, které nelze posoudit z pohledu programátora. Stejně tak je potřeba ověřit, zda vše správně funguje. Nejen uživatelé, kteří pracují s aplikací poprvé, mohou stejného cíle dosáhnout různými způsoby než autor aplikace. Některé postupy při dosahování cílu mu mohou připadat nelogické, přitom je může používat většina uživatelů. Programátor totiž ví, jak jeho aplikace pracuje a jak ji použít. Uživatel se musí nejdříve naučit pracovat s touto aplikací. Je pochopitelné, že větší úspěch mají aplikace, u kterých se nemusí uživatel učit nic komplikované a systém je převážně intuitivní. Testováním lze také získat nápady a připomínky k aplikaci. Navíc je aplikace ohodnocena od uživatelů, kteří ji mohou sami jednou používat.

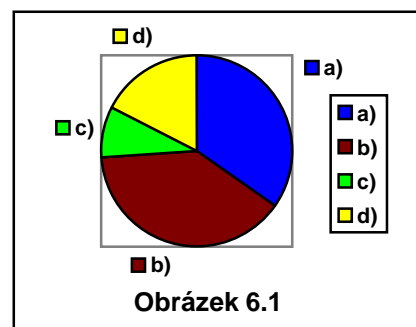
První test zjišťoval počet kliků a přechodů při různých uživatelských akcích. Uživatel by měl mít vše na dosah dvou kliknutí, vždy to ale není zcela možné. Proto by měl systém aspoň jednoduše uživatele navést požadovaným směrem. Pro přechod na stránku s přidáním zvířete bylo potřeba jen dvou kliknutí, toto nové uživatele zajímalo nejvíce. Při vytváření nového zvířete bylo potřeba přejít třemi stranami. Pro detail zvířete stačily uživatelům opět dva kliky. Pro vyhledávání nového přítele bylo potřeba přejít do záložky Přátelé, kde uživatel musel zadat hledaný výraz. Pak se mu objevil seznam, tam si uživatel mohl hledaného člověka přidat nebo dalším kliknutím prohlédnout detail. Pokud uživatel klikl na ikonu Zprávy, zobrazily se mu všechny příchozí. Pro zobrazení odeslaných musel učinit další kliknutí navíc. To samé platilo i pro odeslání nové zprávy.

Druhý test podstoupilo dvacet uživatelů, ti otestovali aplikaci a odpověděli na otázky v dotazníku.

Vyhodnocení dotazníku

1) Jak často využíváte počítač a internet?(Obrázek 6.1)

a) Pracuji s počítačem a Internetem denně.	8	40%
b) Často používám počítač i Internet.	9	45%
c) Méně často používám počítač a Internet.	1	10%
d) Skoro počítač a Internet nepoužívám.	2	20%



První otázka měl za úkol zjistit, jak jsou uživatelé zblběhlí v používání počítače a Internetu. Aplikace klade důraz na jednoduchost, aby byla přístupná uživatelům různých věkových skupin.

2) Jak se Vám líbí vzhled aplikace? (Obrázek 6.2)

a) Líbí se mi.	5	25%
b) Celkem se mi to líbí.	8	40%
c) Chce to vylepšit.	4	20%

d) Nelíbí se mi..... 3 15%

Je důležité budoucí uživatele zaujmout a hlavně neodradit. Proto druhá otázka měla zjistit spokojenost uživatelů.

3) Zdála se Vám práce s aplikací intuitivní? (Obrázek 6.3)

a) Ano, věděl/a jsem, jak aplikaci používat..... 6 30%

b) Spíše ano, některými kroky jsem si nebyl/a jistý/á..... 10 50%

c) Spíše ne, často jsem se ztratil/a a nevěděl/a co dělat..... 3 15%

d) Ne vůbec, nevěděl/a jsem si s aplikací rady. 1 5%

Otázka třetí se snaží zjistit, zda je aplikace opravdu intuitivní a zda není příliš komplikovaná na používání.

4) Jak často jste nevěděli, jak pokračovat dál? (Obrázek 6.4)

a) Často, musel/a jsem se i občas zeptat, jak dál. 1 5%

b) Často jsem váhal/a..... 4 20%

c) Občas jsem zaváhal/a, ale většinou jsem si věděl rady..... 12 60%

d) Vždy jsem věděl/a jak dál..... 3 15%

Čtvrtá otázka se zabývá složitostí systému. Některé požadované akce nemohou být hned přístupné a je nutné dostat se k nim pomocí přechodů přes různé stránky.

5) Podařilo se Vám přidat požadované přátele? (Obrázek 6.5)

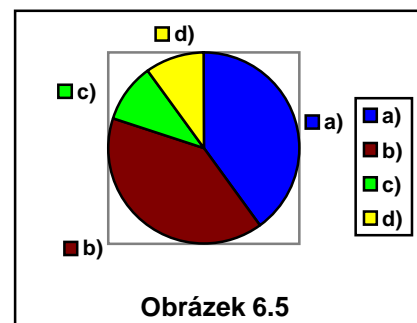
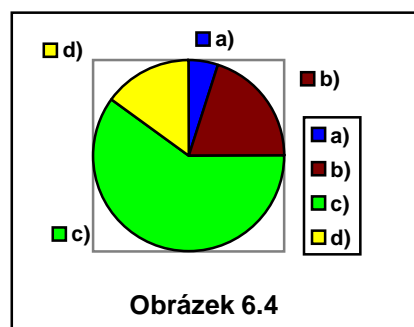
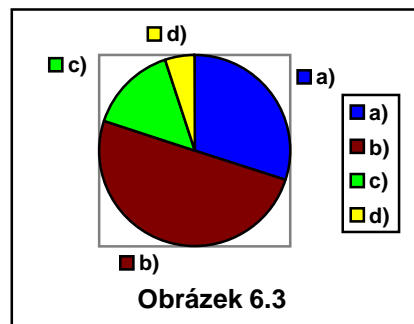
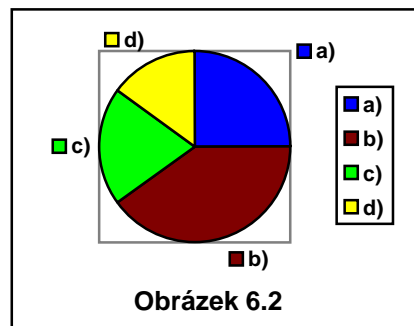
a) Ano, neměl/a jsem s tím problém..... 8 40%

b) Návrh přátelství jsem přijala, ale nevěděl/a jsem, jak přátelství navrhnout. 8 40%

c) Nevěděl jsem, jak přijmout přátelství, navrhnout nové přátelství mi nedělalo problém. 2 10%

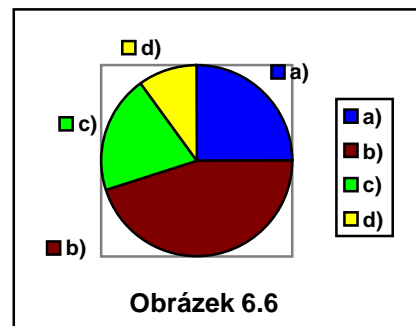
d) Ne, dělalo mi to velké problémy. 2 10%

Nejsložitějším úkonem celého systému je přidání správné osoby, kterou uživatel žádá o přátelství. Uživatel může požádat o přátelství jiného uživatele při vyhledávání a nebo v zobrazeném detailu, aby si byl jistý zda jde o správnou osobu..



6) Bylo přidávání více zvířat zdlouhavé? (Obrázek 6.6)

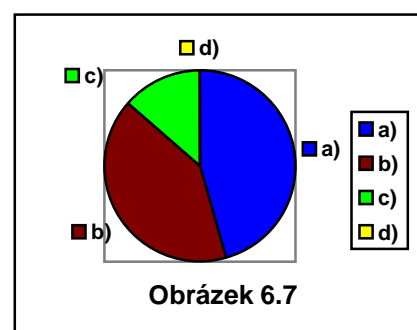
a) Ano, trvalo to dlouho.....	5	25%
b) Spíše ano.....	9	45%
c) Spíše ne.....	4	20%
d) Ne, zdálo se mi to normální.....	2	10%



Při přidávání nového jedince do systému je potřeba zadat jméno a plemeno, v dalším kroku je potřeba vybrat nebo zadat otce jedince. Potom se dostanete k dalšímu kroku, kdy se vybírá nebo zadává matka jedince. Tato otázka se snaží zjistit, zda to uživatelům nebude příliš vadit.

7) Líbí se Vám možnost posílat ostatním uživatelům zprávy? (Obrázek 6.7)

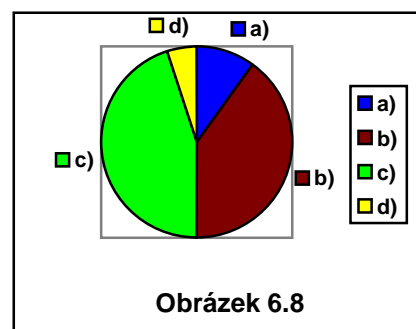
1) Ano, líbilo se mi to, je to užitečné.....	10	50%
2) Spíše se mi to líbilo, využiji to spíše jen občas.....	7	45%
3) Spíše se mi to nelíbilo, asi to vůbec nevyužiji.....	3	15%
4) Ne, nelíbilo se mi to, přijde mi to zbytečné.....	0	0%



Otázka sedmá se zajímá, zda je u uživatelů oblíbená možnost posílat si navzájem zprávy a zda se jedná o zjednodušení a zpříjemnění práce s aplikací. Ta se snaží, aby tato možnost příliš uživatele nerušila.

8) Jste celkově spokojeni s aplikací? (Obrázek 6.8)

a) Jsem spokojen/a, aplikace se mi líbí.....	2	10%
b) Jsem převážně spokojen/a, pár věcí by se mohlo vylepšit.....	8	40%
c) Jsem spíše nespokojen, je potřeba hodně věcí vylepšit.....	9	45%
d) Nejsem vůbec spokojen/a, aplikace se mi nelíbí.....	1	5%



Otázkou se snažím zjistit, zda se aplikace zdařila, zda bude u uživatelů oblíbená.

Podle výsledků dotazníku byli uživatelé spíše spokojeni. Uživatelé dovedli s aplikací pracovat. I když většina uživatelů uvedla, že se jim aplikace líbí, nakonec by na ní hodně věcí vylepšili. Nejvíce se jim líbila možnost posílat ostatním uživatelům zprávy. Přidávání zvířat tvořilo testovací skupině problém, protože se jim to zdálo zdlouhavé. Mezi složitější úkony patřilo přidávání zvířat, ale uživatelé to překvapivě zvládli. Problém jim hlavně dělalo přidání přátel.

7 Závěr

Tato práce popisuje návrh a tvorbu informačního systému pro chovatele. Tato webová aplikace umožňuje uživatelům editovat svá zvířata a pomáhá navazovat přátelství s ostatními chovateli. Chovatelé se mohou slučovat do různých skupin, které si mohou sami vytvořit. Mohou se dozvědět o různých akcích. V rámci tohoto návrhu jsem se seznámila s principy chovu a šlechtění zvířat. Pro implementaci této aplikace bylo potřeba nastudovat několik nových metod a technik. Systém by měl být pro uživatele snadno ovladatelný a intuitivní, aby uživatelé neztratili brzy o tento systém zájem.

Tuto práci jsem si vybrala, protože se mi líbila možnost vytvořit informační systém s tematikou zvířat. Je to dobrá zkušenost pro případnou budoucí praxi. Práce na této bakalářské práci mě zaujala, díky ní jsem získala nové znalosti v oblasti J2EE JSP. Značně jsem si prohloubila znalosti v HTML, CSS, JavaScriptu a celkové tvorby webových aplikací pomocí JSP. Oproti původním očekáváním nejvíce času zabralo studium a implementace AJAXu. Práce s ním ze začátku byla komplikovanější.

Pro úspěšnost webové aplikace je nutné ji neustále inovovat. Tato aplikace pracuje podle předem zadaných kritérií, přesto zde spousta užitečných věcí chybí. Možností rozšíření tohoto systému je hodně. Jedno z vhodných rozšíření v této aplikaci by bylo vytvoření galerie obrázků k jednotlivým zvířatům. Také by bylo vhodné přidat možnost editace profilových obrázků, kdy by si uživatelé mohli zvolit jak velký má být obrázek nebo si zvolit pouze určitý výřez obrázku. Dalším vhodným vylepšením by byla možnost, kdy by si uživatel mohl rozdělit přátele podle určitých kritérií do různých kategorií.

Literatura

- [1] Jan Jelínek, Vladimír Ticháček a kolektiv: *Biologie pro gymnázia*, Nakladatelství Olomouc, Olomouc, 2000, 4. rozšířené vydání, ISBN: 80-7182-107-1
- [2] Jaromír Dostál: *Genetika a šlechtění plemen psů*, Dona, České Budějovice, 2007, ISBN: 978-80-7322-104-1
- [3] WWW stránky: Internetový holubník: Vše o holubech, stránky o chovu a šlechtění holubů [online]
<http://www.holubi.wgz.cz/> [cit. 2010-02-15]
- [4] WWW stránky: Zootechnika a chov hospodářských zvířat [online]
<http://www.zootechnika.estranky.cz/clanky/chov-skotu> [cit. 2009-12-15]
- [5] WWW stránky: CZ – Pes: stránky plné psů. Seznam uznaných plemen podle FCI [online]
http://www.cz-pes.cz/atlas_index.php [cit. 2009-12-15]
- [6] WWW stránky: Chovatelství [online]
<http://www.chovatelstvi.cz/> [cit. 2009-12-15]
- [7] WWW stránky: Evidence a rodokmeny poštovních holubů, Vít'a Ištváněk [online]
Aplikace v MS Excel
<http://istvanek.net/holubi/index.html> [cit. 2009-12-15]
- [8] Barry Burd: *JSP: JavaServer Pages: Podrobný průvodce*. Computer Press, Praha, 2003, 1. vydání, ISBN 80-7226-804-X
- [9] Ladislav Lacko: *AJAX: Hotová řešení*. Computer Press, Brno, 2008, 1. vydání, ISBN 978-80-251-2108-5
- [10] Steven M. Schafer: *HTML, XHTML a CSS: bible [pro tvorbu WWW stránek]*. Grada, Praha, 2009, 1. vydání, ISBN 978-80-247-2850-6
- [11] Budi Kurniawan: *Struts 2 Design and Programming: A Tutorial*. BrainySoftware, 2008, ISBN: 978-0-9803316-0-8
- [12] Keniv Yank, Cameron Adams: *Začínáme s JavaScriptem*, Zoner Press, Brno, 2008, 1. vydání, ISBN: 978-80-86815-94-7
- [13] Dave Thau: *Velký průvodce JavaScriptem: tvorba interaktivních webových stránek v praxi*. Grada, Praha, 2009, 1. vydání, ISBN: 978-80-247-2211-5
- [14] Steve Krug: *Web design: nenutíte uživatele přemýšlet*. Computer Press, Brno, 2006, 2. aktualizované vydání, ISBN: 80-251-1291-8

- [15] Hana Kanisová, Miroslav Müller: *UML srozumitelně*. Computer Press, Brno, 2006, 2. aktualizované vydání, ISBN: 80-251-1083-4
- [16] WWW stránka: Interval.cz: JavaServer Pages pro všechny [online].
<http://interval.cz/clanky/jaserver-pages-pro-vsechny/> [cit. 2010-03-20]
- [17] WWW stránky: Sun Developer Network, J2EE – JavaServer Pages Technology, Oficiální stránky [online].
<http://java.sun.com/products/jsp/?intcmp=3184> [cit. 2010-01-18]
- [18] WWW stránky: Struts1 vs Struts 2 [online].
<http://www.roseindia.net/struts/struts1-vs-struts2.shtml> [cit. 2010-05-01]
- [19] WWW stránky: The Apache Software Foundation: Struts 2.x [online].
<http://struts.apache.org/2.x/> [cit. 2010-2-06]
- [20] Radim Hradecký: Srovnání softwarových frameworků, Praha, 2009 [online].
<http://www.hradecky.org/wp-content/uploads/Hradecky-diplomka.pdf> [cit. 2010-05-01]
- [21] WWW stránky: Wikipedie
http://cs.wikipedia.org/wiki/Hlavn%C3%AD_strana [cit. 2009-12-20]
- [22] WWW stránky: AJAX DOJO
<http://www.dojotoolkit.org/> [cit. 2010-04-15]
- [23] WWW stránky: Jak psát web: o tvorbě internetových stránek
<http://www.jakpsatweb.cz/> [cit. 2010-04-15]
- [24] Radim Reš: *Evidence rodokmenů plemenných chovů*, Brno, 2010, bakalářská práce FIT VUT v Brně

Seznam příloh

Příloha 1: Obsah přiloženého CD

Příloha 2: Seznam plemen psů uznaných FCI

Příloha 3: ER diagram

Příloha 1

Obsah přiloženého CD

- složka **Technická zpráva** – obsahuje text bakalářské práce ve formátu PDF a DOC

 System pro evidenci chovů.pdf

 System pro evidenci chovů.doc

- složka **Dokumentace** – obsahuje programovou dokumentaci a manuál k aplikaci

 manual.pdf

 Javadoc

- složka **Aplikace** – obsahuje zdrojové soubory aplikace ve formě NetBeans projektu

- readme.txt – je návod k použití CD

Příloha 2

Seznam plemen psů uznaných FCI

Zdroj: http://www.cz-pes.cz/atlas_index.php a http://cs.wikipedia.org/wiki/Plemena_ps%C5%AF

Číslo skupiny	Skupina	Typy plemen
Skupina 1	Ovčáci a honáčí psi	Ovčácká a pastevecká plemena Australský honácký pes, Australská kelpie, Australský ovčák, Bearded kolie, Beauceron, Belgický ovčák (Groenendael, Laekenois, Malinois, Tervueren), Bergamský ovčák, Bílá kolie, Bílý švýcarský ovčák, Bobtail, Border kolie, Briard, Portugalský ovčák, Československý vlčák, Holandský ovčák, Holandský ovčácký pudl - šápendús, Chorvatský ovčák, Jihoruský ovčák, Katalánský ovčák, Kolie dlouhosrstá, Kolie krátkosrstá, Komondor, Kuvasz, Německý ovčák, Malorský ovčák, Maremmansko-abruzský pastevecký pes, Mudi, Pikardský ovčák, Podhalaňský ovčák, Polský ovčák nížinný, Puli, Pumi, Pyrenejský ovčák, Saarloosův vlčák, Sheltie, Slovenský čuvač, Šiperka, Welsh Corgi Cardigan, Welsh Corgi Pembroke Honácká plemena Australský honácký pes, Ardenský bouvier, Flanderský bouvier
Skupina 2	Pinčové a knírači, molosoidní	Pinčové a knírači Černý teriér, Dánsko-švédský selský pes, Dobrman, Holandský pinč, Knírač velký, Knírač střední, Knírač malý, Malý krátkosrstý pinč, Německý pinč, Opičí pinč, Rakouský pinč, Trpasličí pinč Molosoidní plemena (dogovití psi) Molosoidní plemena typu mastila Anglický buldok, Anglický mastif, Argentinská doga, Bordeauxská doga, Brazilská fila, Bullmastif, Dánská doga - broholmer, Italský cane corso, Kanárská doga, Malorská doga, Neapolský mastin, Německý boxer, Německá doga, Kanárská doga, Malorská doga, Šarpej, Rotvajler, Tosa-Inu Molosoidní plemena horského typu Aidi, Anatolský pastevecký pes, Bernardýn, Cao de Castro Laboreiro,

		<p>Estrelský pastevecký pes, Hovawart, Kavkazský pastevecký pes, Krašský pastevecký pes, Landseer, Leonberger, Novofundlandský pes, Pyrenejský horský pes, Pyrenejský mastin, Rafeiro do Alentejo, Středoasijský pastevecký pes, Šarplaninský pastevecký pes, Španělský mastin, Tibetská doga</p> <p>Švýcarští salašnickí psi</p> <p>Appenzellský salašnický pes, Bernský salašnický pes, Etenbuchský salašnický pes, Velký švýcarský salašnický pes</p>
Skupina 3	Teriéři	<p>Velcí a střední teriéři</p> <p>Airedale teriér, Bedlington teriér, Border teriér, Brazílský teriér, Foxteriér drsnosrstý, Foxteriér hladkosrstý, Glen of imaal teriér, Irský glen of Imaal teriér, Irský teriér, Irish soft coated wheaten teriér, Kerry blue teriér, Lakeland teriér, Manchester teriér, Německý lovecký (jagd) teriér, Parson Russel teriér, Soft coated wheaten teriér, Welsh teriér</p> <p>Malí teriéři</p> <p>Australský teriér, Cairn teriér, Český teriér, Dandie dinmont teriér, Japonský teriér, Jack Russell teriér, Jagd teriér, Parson Russell teriér, Norfolk teriér, Norwich teriér, Sealyham teriér, Silky teriér, Skotský teriér, Skye teriér, West highland white teriér</p> <p>Teriéři typu bull</p> <p>Americký stafordširský teriér, Bulteriér, Miniaturní bulteriér, Stafordširský bulteriér</p> <p>Toy teriéři</p> <p>Anglický toy teriér, Australský silky teriér, Yorkšírský teriér</p>
Skupina 4	Jezevčíci	<p>Jezevčík drsnosrstý standardní, Jezevčík hladkosrstý standardní, Jezevčík dlouhosrstý standardní, Jezevčík drsnosrstý trpasličí, Jezevčík hladkosrstý trpasličí, Jezevčík dlouhosrstý trpasličí, Jezevčík drsnosrstý králičí, Jezevčík hladkosrstý králičí, Jezevčík dlouhosrstý králičí</p>
Skupina 5	Špicové a tzv. primitivní typy	<p>Severští sáňoví psi</p> <p>Aljašský malamut, Grónský pes, Samojed, Sibiřský husky</p> <p>Severští lovečtí psi</p> <p>Finský špic, Jaemthund, Karelský medvěd pes, Norbotenský špic, Norský losí pes černý, Norský losí pes šedý, Norský lundehund, Ruskoevropská lajka, Východosibiřská lajka, Západosibiřská lajka</p>

		<p>Severští hlídací a ovčáčtí psi</p> <p>Islandský pes, Lapinkoira, Lapinporokoirra, Laponský pes, Norský buhund, Švédský lapphund, Vaestgoetlandský špic - vallhund</p> <p>Evropští špicové</p> <p>Italský volpino, Německý špic</p> <p>Asijští špicové a příbuzná plemena</p> <p>Akita-Inu, Americká akita, Čau-čau, Eurasier, Hokkaido-Ken, Japonský špic, Kai-Inu, Kishu-Inu, Korejský Jindo Dog, Shiba-Inu, Shikoku-Inu</p> <p>Primitivní plemena</p> <p>Basenji, Canaanský pes, Faraónský pes, Mexický naháč, Peruánský naháč</p> <p>Primitivní lovecká plemena</p> <p>Faraonský chrt, Ibizský podenco, Kanárský podenco, Portugalský podengo, Sicilský chrt</p> <p>Primitivní lovecká plemena s protisměrně rostoucí srstí na hřbetě (ridgem)</p> <p>Thajský ridgeback</p>
Skupina 6	Honiči, barváři a příbuzná plemena	<p>Honiči</p> <p>Velcí honiči</p> <p>Americký foxhound, Anglický foxhound, Ariégoiský honič, Billy, Black and tan coonhound, Bloodhound, Francouzský bíločerný honič, Francouzský bílooranžový honič, Francouzský trikolorní honič, Poitevin, Velký anglo-francouzský bílo-černý honič, Velký anglo-francouzský bílo-oranžový honič, Velký anglo-francouzský trikolorní honič, Velký gaskoňsko-saintongeiský honič, Velký modrý gaskoňský honič, Velký griffon vendéenský honič, Vydrař (otterhound)</p> <p>Střední honiči</p> <p>Anglo-francouzský honič de Petite Venerie, Ariégois, Artoiský honič, Bígl harrier, Bosenský hrubosrstý honič, Brandlův brakýř, Černohorský planinský honič, Dunker, Finský honič, Haldenův honič, Hamiltonův honič, Harrier, Hygenův honič, Hrubosrstý vendéský honič - Briquet griffon vendéen, Istrijský hrubosrstý honič, Istrijský krátkosrstý honič, Italský segugio, Malý gaskoňsko-saintongeiský honič, Malý modrý gaskoňský honič, Modrý gaskoňský honič, Nivernaisský (hrubosrstý) honič, Plavý bretaňský hrubosrstý honič, Polský honič, Porcelaine, Posávský honič, Řecký honič, Sedmihradský honič, Schillerův honič,</p>

		<p>Slovenský kopov, Smalandský honič, Srbský honič, Srbský trikolorní honič, Španělský sabueso, Štýrský brakýř, Švýcarský honič - bernský, luzernský, jurský, schwyzský honič, Tyrolský honič</p> <p>Malí honiči</p> <p>Artésko-normandský basset, Artoiský basset, Baset, Bígl, Drever, Malý hrubosrstý vendéský basset, Modrý gaskoňský basset, Německý honič, Plavý bretaňský basset, Švýcarský nízkonohý honič - bernský, luzernský honič, jurský honič, schwyzský honič</p> <p>Nízkonohý honič</p> <p>Velký hrubosrstý vendéský basset, Vestfálský jezevčikovitý honič, Bígl-Harrier</p> <p>Barváři</p> <p>Alpský jezevčikovitý brakýř, Bavorský barvář, Hannoverský barvář</p> <p>Příbuzná plemena</p> <p>Dalmatin, Rhodéský ridgeback</p>
Skupina 7	Ohaři	<p>Kontinentální ohaři</p> <p>Typu brague (krátkosrstí)</p> <p>Ariegský ohař, Auvergneský ohař, Burbonský ohař, Dupuyův krátkosrstý ohař, Burgoský perdiquero, Francouzský ohař gaskoňského typu, Francouzský ohař pyrenejského typu, Italský ohař, Maďarský ohař drátosrstý, Maďarský ohař hladkosrstý, Maďarský ohař krátkosrstý, Německý ohař drátosrstý, Německý ohař krátkosrstý, Německý ohař ostnosrstý, Portugalský perdiguero, Pudlpointr, Saint-Germainský ohař, Starodánský ohař, Výmarský ohař</p> <p>Typu španěl (dlouhosrstí)</p> <p>Bretaňský ohař, Drentsche koroptvář, Francouzský dlouhosrstý ohař, Malý münsterlandský ohař, Modrý pikardský ohař, Německý ohař dlouhosrstý, Ohař z Pont-Audemer, Pikardský ohař, Pont-Odemerský ohař, Stabyhoun (fríský ohař), Velký münsterlandský ohař</p> <p>Typu griffon (hrubosrstí)</p> <p>Český fousek, Italský spinone, Spinone, Korthalsův grifon, Slovenský ohař hrubosrstý</p> <p>Ohaři britských ostrovů</p> <p>Anglický setr, Gordonsetr, Irský červenobílý setr, Irský setr, Pointer</p>
Skupina 8	Retrívři, slidiči a vodní psi	<p>Retrívři</p> <p>Curly coated retrívř, Flat coated retrívř, Chesapeake Bay retrívř, Labradorský retrívř, Nova Scotia duck tolling retrívř, Zlatý retrívř</p>

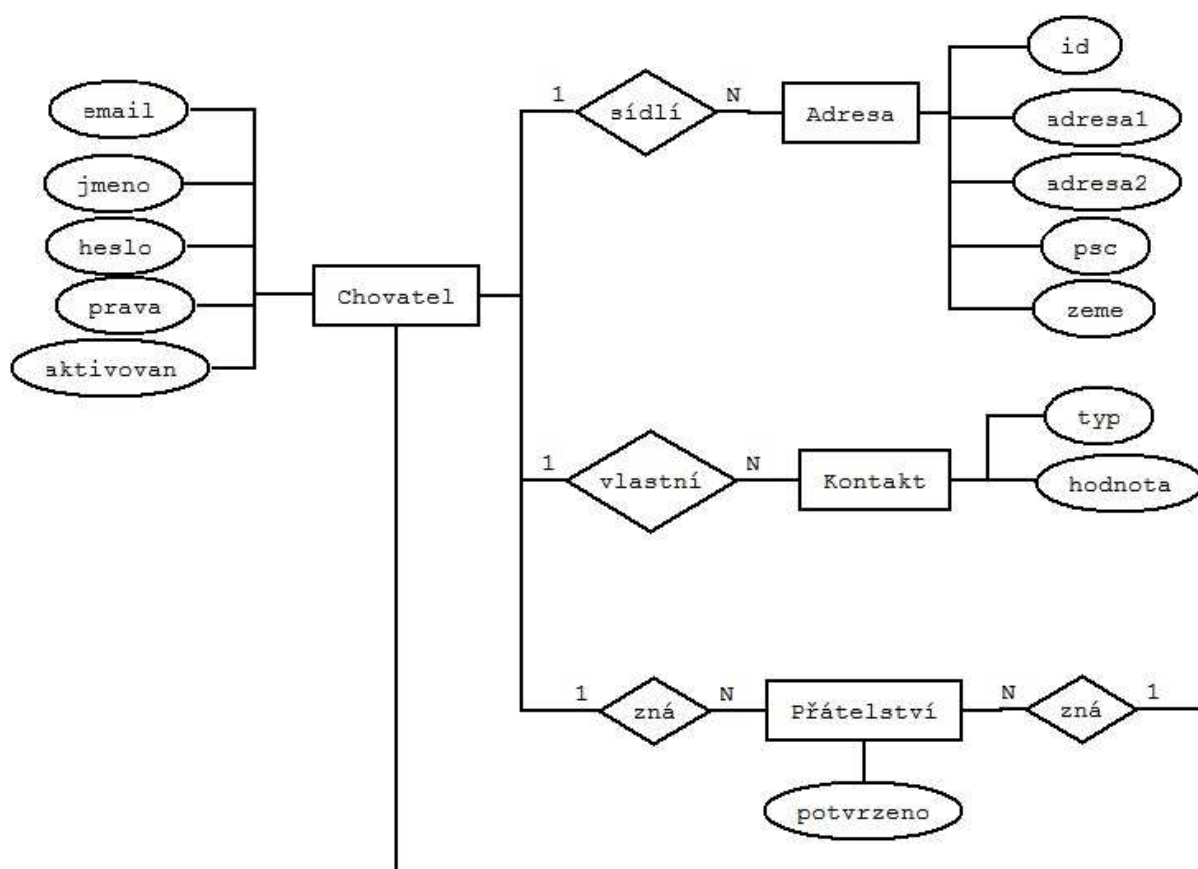
		<p>Slídiči</p> <p>Americký kokršpaněl, Anglický kokršpaněl, Anglický špringršpaněl, Clumber španěl, Field španěl, Kooikerhondje, Sussex španěl, Německý křepelák, Welššpringršpaněl</p> <p>Psi vodní</p> <p>Americký vodní španěl, Barbet, Irský vodní španěl, Portugalský vodní pes, Španělský vodní pes, Wetterhound</p>
Skupina 9	Společensk á plemena a toy	<p>Bišonci a příbuzná plemena</p> <p>Boloňský psík, Bišonek, Coton de Tuléar, Havanský psík, Lviček, Maltézský psík</p> <p>Pudl - Střední pudl, Toy pudl, Trpasličí pudl, Velký pudl</p> <p>Papillon - Papillon</p> <p>Belgičtí psíci - Belgický grifonek, Brabantík, Bruselský grifonek</p> <p>Naháči - Čínský chocholatý pes</p> <p>Tibetská plemena - Lhasa Apso, Shih-tzu, Tibetský španěl, Tibetský teriér</p> <p>Čivava - Čivava</p> <p>Angličtí společenští španělé a ruský toy španěl</p> <p>Kavalír King Charles španěl, King Charles španěl, Ruský toy španěl</p> <p>Japan-chin a pekingský palácový psík</p> <p>Japan-chin, Pekingský palácový psík</p> <p>Kontinentální společenští španělé - Malý kontinentální španěl</p> <p>Kromfohrländer - Kromfohrländer</p> <p>Malí molossoidní psi - Bostonský teriér, Francouzský buldoček, Mops</p>
Skupina 10	Chrti	<p>Dlouhosrstí chrti</p> <p>Afgánský chrt, Barzoj, Saluki (Perský chrt)</p> <p>Hrubosrstí chrti</p> <p>Irský vlkodav, Skotský jelení pes (deerhound)</p> <p>Krátkosrstí chrti</p> <p>Anglický chrt (greyhound), Azavak, Italský chrtík, Maďarský chrt, Polský chrt, Sloughi, Španělský galgo, Vipet</p>

Příloha 3

ER diagram

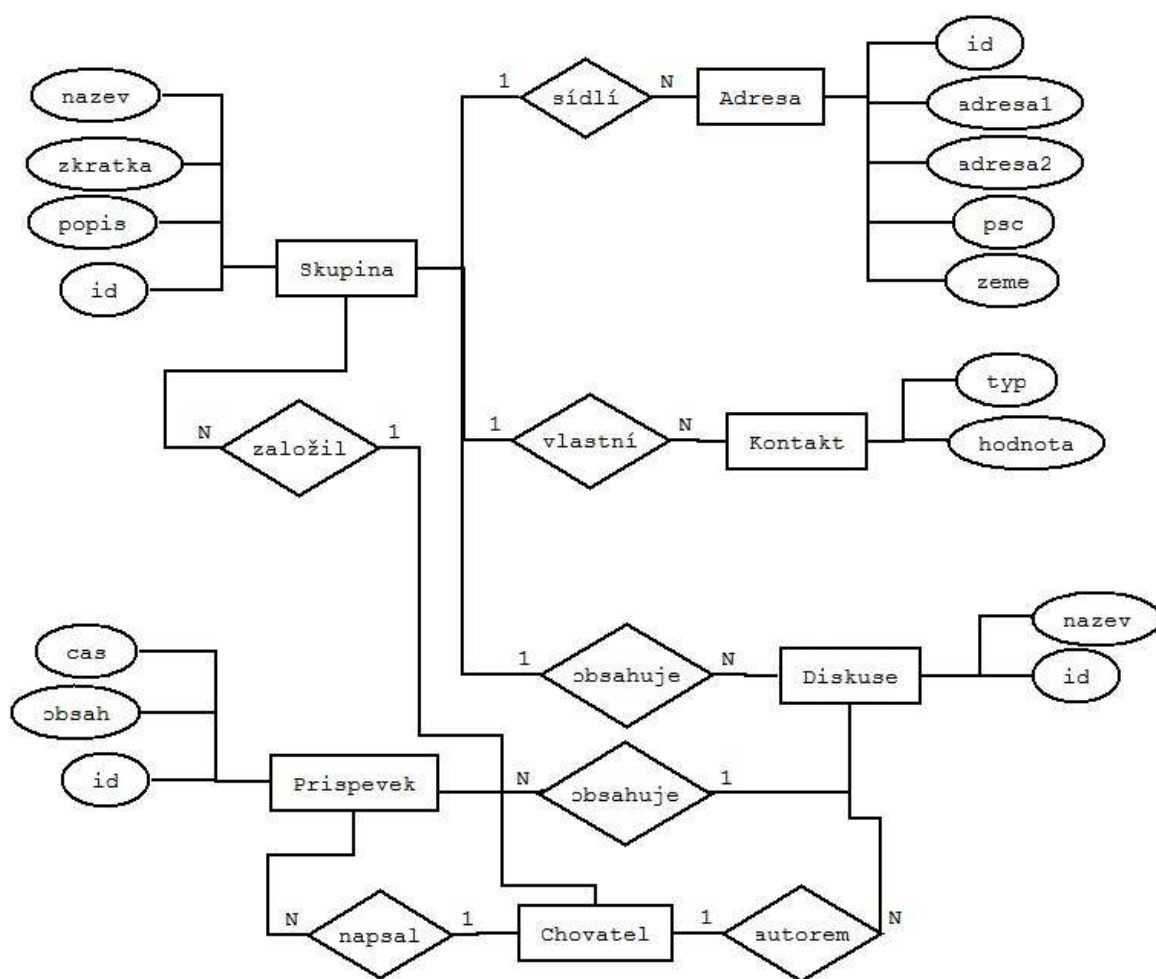
ER- diagram, zobrazený na **obrázcích 1-5**, je schématem relační databáze pro tuto webovou aplikaci. Základními entitami jsou *Chovatel*, *Adresa*, *Kontakt*, *Přátelství*, *Skupina*, *Zpráva* a *Soutěže*.

Chovatel může mít více adres a více kontaktů (**Obrázek 1**). *Chovatel* má atributy *email*, *jmeno*, *heslo*, *prava* a *aktivovan*. *Chovatel* má vztah – *sídlí* s entitou *Adresa* a vztah – *vlastní* s entitou *Kontakt*. *Přátelství* s ostatními chovateli má atribut *Potvrzeno*.



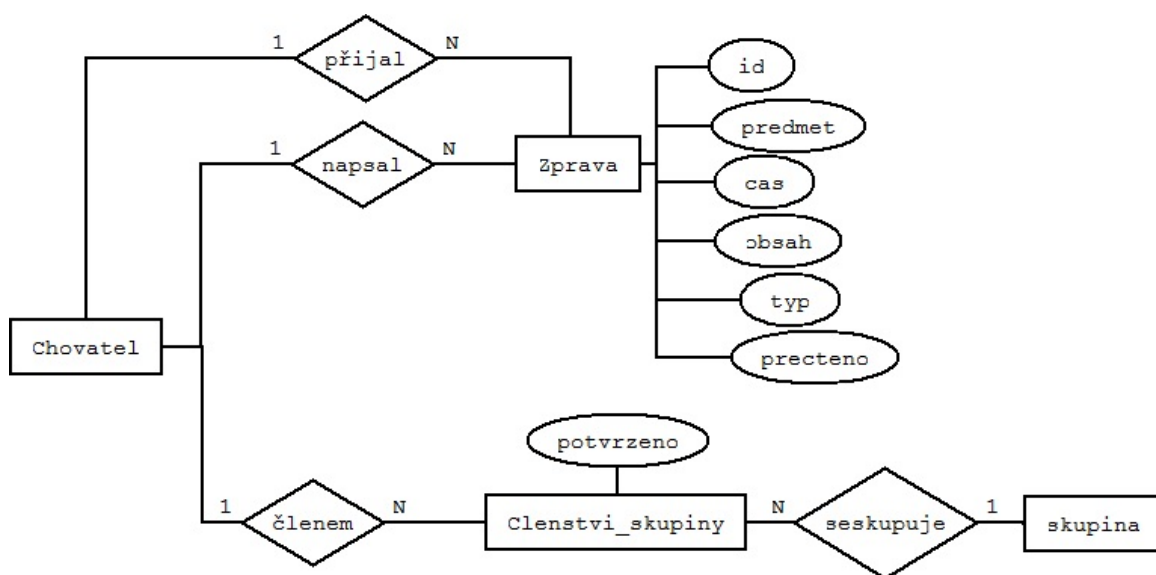
Obrázek 1: ER diagram - Chovatel

Entita *Skupina* má atributy *nazev*, *zkratka*, *popis* a *id* (**Obrázek 2**). Entita *Skupina* má vztahy – *sídlí*, *vlastní* a *obsahuje* s entitou *Adresa*, *Kontakt* a entitou *Diskuse*. *Diskuse* má autora, kterým je *Chovatel*. *Chovatel* může psát příspěvky, ty mají atribut *cas*, *obsah*, *id*.



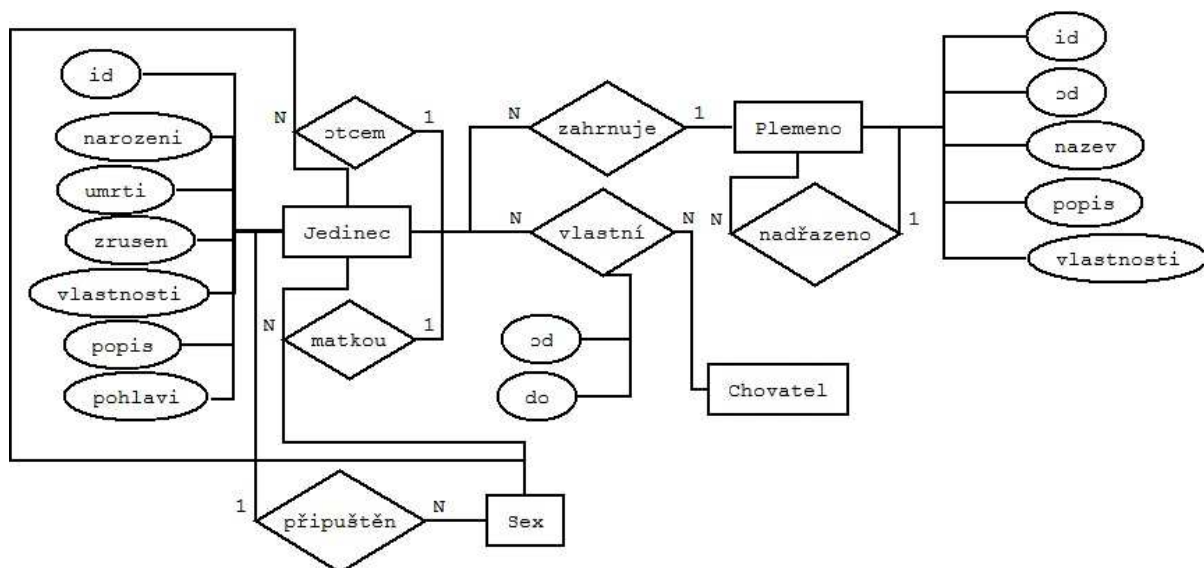
Obrázek 2: ER diagram- Skupina

Entita Zprava může mít dva autory – Chovatel, Clenstvi_skupiny (Obrázek 3). Zprava má atributy id, predmet, cas, obsah, typ a precteno. Chovatel musí být členem Skupiny, aby mohl za skupinu poslat zprávu.



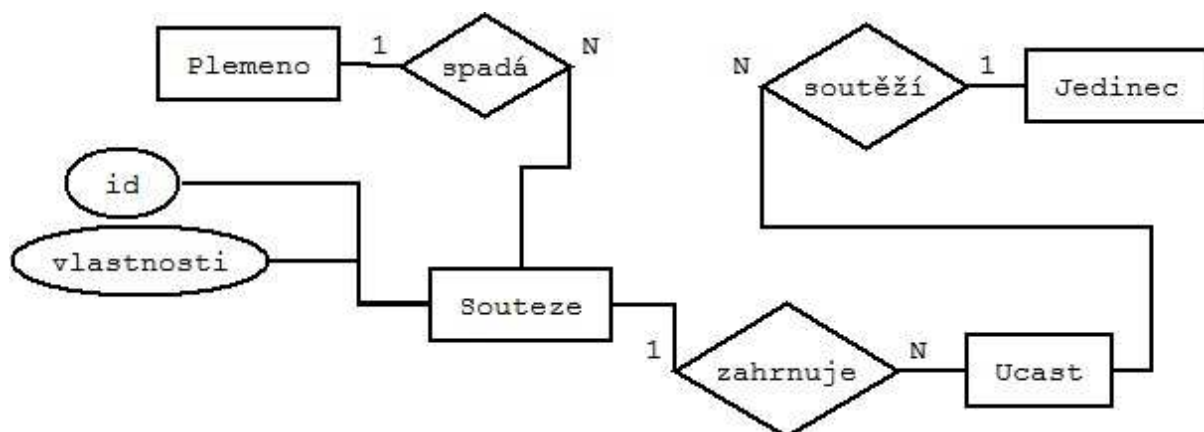
Obrázek 3: ER diagram- Zprava

Entita *Jedinec* má atributy *id*, *narozeni*, *umrti*, *zrusen*, *vlastnosti*, *popis* a *pohlavi* (**Obrázek 4**). *Jedinec* je připouštěn, což znázorňuje vztah entit *Jedinec* a *Sex*. *Chovatel* vlastní jedince v určitém časovém úseku (atributy *od*, *do*). *Jedinec* je konkrétního plemene. Entita *Plemeno* má atributy *id*, *od*, *nazev*, *popis* a *vlastnosti*. *Plemeno* znázorňuje celou zvířecí říši i jednotlivé druhy a plemena.



Obrázek 4: ER diagram - Jedinec

Entita *Soutez* je ve vztahu – *spadá* s entitou *Plemeno* (**Obrázek 5**). Soutěží se účastní jedinci, což znázorňuje vztah *soutěží*. Soutěže mají *vlastnosti* a *id*.



Obrázek 5: ER diagram - Soutěže